Platformode Payment
Integration Specification

Document Amendment Register

The enclosed document has been amended by the following description:

Version	Date	Description
1.0.0	23-Sep-2019	Initial draft
1.1.0	25-Sep-2019	Added WooCommerce plugin installment instruction
2.0.0	19-Nov-2019	Added white label payment solution
2.10.1	08-Nov-2020	New parameter added to Get Installment API
2.10.2	16-Nov-2020	New status code added at order status API
2.10.3	30-Nov-2020	New request parameter added at Order Status API
2.10.4	01-Dec-2020	New Request Parameter is added with paySmart3d API
2.10.5	07-Dec-2020	Merchant Commission API
2.10.6	10-Dec-2020	Response validation using hash_key
2.10.7	11-Dec-2020	ip parameter is added with 2d/3d payment request.

2.10.8	23-Dec-2020	New Request Parameter is added with 2d payment API
2.10.9	23-Dec-2020	paySmart2d API implemented
2.10.10	01-Jan-2021	New response param ‘expires_at’ added to get token api
2.10.11	01-Feb-2021	New error codes and messages added
2.10.12	01-March-2021	Pay With `Card Token`
2.10.17	12-April-2021	Card Token/ADD/EDIT/DELETE/VIEW API added
2.10.18	20-April-2021	Get Transactions by Date API added
2.10.21	24-May-2021	
2.10.23	16-June-2021	Added new parameters on Refund API
2.10.24	23-June-2021	Pay With `Card Token`
2.10.25	29-June-2021	Update Response in payment
2.10.26	13-July-2021	Added Response Explanation at payment API
2.10.27	02-August-2021	Refund web hook
2.10.28	19-August-2021	New error code added
2.10.29	07-September-2021	Pay With `Card Token(Non-Secure)`
2.10.30	17-September-2021	Error code added

2.10.31	04-October-2021	maturity_period, payment_frequency params added in paysmart3d request also i success response of get installment api
2.10.32	22-October-2021	total parameter added to pre auth confirmation api
2.10.34	16-November-2021	card_holder_name set to max = 100,
2.10.35	08-December-2021	New error code and error message added
2.10.36	14-December-2021	Error message changed of error code 113
2.10.37	27-Decemeber-2021	⇨ Added merchant_commission and user_commission key parameters **Check Order Status** and **Sale WebHo** Response ⇨ Rate Limiter : for API Request
2.10.38	21-January-2022	⇨ Sale Webhook: content-type : definition
2.10.39	23-January-2022	Added All Transaction API
2.10.40	08-February-2022	Refund Amount Empty String
2.10.41	06-March-2022	Recurring Payment update api, query api, delete api
2.10.42	12-April-2022	md_status in order query and sale webhook
2.10.43	21-April-2022	Hourly and Range wise transactio filtering system added
2.10.44	21-April-2022	Inclusion of "settlement_date" in checkStatus api

2.10.45	07-September-2022	Merchant 3D Model : CompletePayment API
2.10.46	29-September-2022	Insurance Payment
2.10.47	15-November-2022	CheckStatus response parameters original_bank_error_code and original_bank_error_description
2.10.48	06-December-2022	**response_method** parameter for PaySmart 3D
2.10.49	20-April-2023	GetTransactions Api: New optional parameter included. "order_by_option"
2.10.50	25-April-2023	PayByCardToken API: New optional parameter included. "is_remote_card_token"
2.10.51	21-Jun-2023	VPos Insurance Payment cc_no parameter definition change.
2.10.52	10-Oct-2023	New request parameters added in purchase link api

Table of Contents

Platformode Payment 1

Integration Specification 1

Document Amendment Register 1

Table of Contents 4

1.0 INTRODUCTION 6
1.1 PURPOSE 6
1.2 SCOPE 6
1.3 Access URLs 6
1.4 Rate Limiter 7
2.0 Branded Solution with 2d/3d 7
2.1 Check Order status 15
3. Get Token 18
4. Get Installments 20
7.0 Plugins 24
10.0. Transaction Refund 31
Refund WebHook 34
Refund WebHook Response Validation Using Hash key. 35
11.0. Recurring WebHook 36
11.0.1 Update Recurring 39
11.0.2 Query Recurring Information 41
11.0.3 Delete Recurring 43
11.1. Sale WebHook 45
11.2. Merchant Active Installments 47
11.3. Pay With White Label Smart 3D 48
11.4. Merchant Commission API 55
11.5. Response Validation Using Hash key. 57
11.6. Pay With White Label Smart 2D 58
11.7. Pay With CardToken 65
11.8 SaveCard Api 69
11.9 Save Card Edit Api 72
11.10 SaveCard Delete Api 75
11.11 SaveCard : Get Card Tokens Api 77
11.12 Get Transactions 79
11.13 Payment Confirmation API 86
11.14. Pay With Card Token (Non-Secure) 89
11.15. All Transaction API 95

11.16 Merchant 3D Model : CompletePayment Api 101
12.0 Server Status Codes 104
13.0 Application Status Codes 105

1.0 INTRODUCTION

This document provides a conceptual view of Platformode Payment Integration as well as to aid merchants and developers to better understand the working mechanism of the payment system.

1.1PURPOSE

The purpose of this document is to guide merchants to integrate the Platformode payment system on their website in order to process payment with various payment options.

1.2SCOPE

The scope of the integration between merchant applications and Platformode payment system covers the functional requirement for the payment integration requirement.

1.3Access URLs

The following access URLs will be used based on the environment.

Environment	URL Format
Test Server	https://testapp.platformode.com.tr /ccpayment
Live Server	https://app.Platformode.com.tr/ccpayment

1.4 Rate Limiter

To prevent double click all Happy Flow API(s) have been implemented with 2 second rate limiter, i.e. identical multiple back to back request within 2 seconds will cause an exception.

2.0 Branded Solution with 2d/3d

Platformode admin would decide whether the payment from the merchant website should be done using 2D or 3D. The following request should be sent to the Platformode payment integration system from the merchant website. The CURL post request should be received at Platformode with the following parameters:

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/purchase/link	*application/x-www-form-urlencoded*

Type	Params	Data Type	Condition
KEY	merchant_key	string	Mandatory
KEY	invoice	string	Mandatory
KEY	currency_code	string	Mandatory
KEY	name	string	Mandatory
KEY	surname	string	Mandatory
	bill_address1		

KEY KEY KEY KEY KEY KEY KEY KEY KEY Key Key KEY KEY	bill_address2 bill_city bill_postcode bill_state bill_country bill_email bill_phone sale_web_hook_key ip transaction_type vpos_type identity_number	string string string string string string string string string string string string string	Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional
KEY	is_cvv_less	string	Optional
KEY	installments_number	string	Optional

Request for Recurring

KEY	order_type	*integer*	Mandatory

KEY	recurring_payment_number	*integer*	Mandatory
KEY	recurring_payment_cycle	*string*	Mandatory
KEY	recurring_payment_interval	*integer*	Mandatory
KEY	recurring_web_hook_key	*string*	Mandatory

`merchant_key`

merchant_key is a unique key assigned to the merchant. It must be sent from a merchant website.

`invoicee`

invoice is a json formatted string combined with a list of item name, quantity and unit price etc.

For example, if there are three products,

Product 1#

Name: Item1, Qty: 2, Unit Price: 200

Product 2#

Name: Item2, Qty: 1, Unit Price: 100

Product 3#

Name: Item3, Qty: 2, Unit Price: 400

The invoice string will json string of the following array:

$invoice['invoice_id'] = “345345535”; // One unique id which will be available in the return or cancel URL

$invoice['invoice_description'] = “ INVOICE TEST DESCRIPTION” ;

$invoice['total'] = 1300

$invoice[discount] = 220 //The amount of coupon code or discount value

$invoice[coupon] = “3XY8P” //coupon code in case applicable

$invoice['return_url'] = “https://<your_success_url>”

$invoice['cancel_url'] = “https://<your_fail_or_cancel_url>”

```
$invoice['items'] = array(
        array("name"=>"Item1","price"=>200,"quantity"=>2,"description"=>"item1 description"),
        array("name"=>"Item2","price"=>100,"quantity"=>1,"description"=>"item2 description"),
        array("name"=>"Item3","price"=>400,"quantity"=>2,"description"=>"item3 description"),
);
//billing info
    $invoice['bill_address1'] = 'Address 1'; //should not more than 100 characters
    $invoice['bill_address2'] = 'Address 2'; //should not more than 100 characters
    $invoice['bill_city'] = 'Istanbul';
    $invoice['bill_postcode'] = '1111';
    $invoice['bill_state'] = 'Istanbul';
    $invoice['bill_country'] = 'TURKEY';
    $invoice['bill_phone'] = '008801777711111';
    $invoice['bill_email'] = 'demo@Platformode.com.tr';
$invoice['sale_web_hook_key'] = 'sale_web_hook_key';// This key must be assigned on Platformode Merchant Panel
$invoice['trasaction_type'] = 'Pre-Authorization'; //Optional
```

$invoice['vpos_type'] = 'insurance': only applicable for Pay2D solution to initiate an insurance payment by an insurance Merchant. If presents value must be "insurance", otherwise it will throw a validation exception

$invoice['identity_number'] = '12345678901': only applicable for Pay2d Solution to initiate an insurance payemt by an insurance Merchant when "vpos_type":"insurance" is present. Value (tckn/vkn/tin) must be within 10-11 digits

//**Recurring info**

```
    $invoice['order_type'] = 1; //order type 1 for recurring payment
    $invoice[' recurring_payment_number'] = 2; //must be integer value
   $invoice[' recurring_payment_cycle'] = 'M'; // e.g:  D, M, Y
```

$invoice[' recurring_payment_interval'] = 2; //must be integer value

$invoice[' recurring_web_hook_key'] = 'key_name'; //This key must be assigned in the Platformode merchant panel.

All taxes and shipping charges will be added as items in the invoice items array. The item names should be "Tax" and "Shipping Charge" respectively with quantity 1.

Final Request Pattern:

```
$invoice = json_encode($invoice);

$post = array(
    'merchant_key' => $merchant_key,
    'invoice' => $invoice,
    'currency_code' =>  $currency,
    'name' => $name,
    'surname' => $surname
);
```

currency_code

currency_code is code of currency. For example USD, TRY, EUR etc.

name

name First name of the person. For example, if the name of the person who is buying the product is "john Dao", then name should be "john"

surname

surname Last name of the person. For example, if the name of the person who is buying the product is "john Dao", then surname should be "Dao"

sale_web_hook_key

*sale_web_hook_key*is an optional key. When a purchase request is completed, Platformode sends a post request. So that merchant can perform an event on their site. Platformode validates that this key must exist in the database. Merchant must assign a Sale web hook URL on the Platformode Merchant Panel against this key.

transaction_type

For *transaction_type* "PreAuth", a pending transaction is created in the system. But later it is converted to Completed by the merchant's confirmation.

order_type

If *order_type=1*, Platformode validates payment for recurring. Then *recurring_payment_number, recurring_payment_cycle, recurring_payment_interval*keys should not be empty.

recurring_payment_number

*recurring_payment_number*defines installment count. If first_amount is $100 and **recurring_payment_number**is 5, then the total amount will be deducted as $100*5 = $500. (Cost of transaction may be added with each transaction)

recurring_payment_cycle

recurring_payment_cycle defines the unit type of *recurring_payment_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**:Years

recurring_payment_interval

*recurring_payment_interval*defines interval value. If *recurring_payment_interval*= 2 and *recurring_payment_cycle* = "M" then transaction will occur once in every 2 Months.

recurring_web_hook_key

*recurring_web_hook_key*defines merchant recurring web hook url . An URL must be assigned on the Platformode Merchant Panel against this key. Platformode validates this key must exist in the database and it is a required value when payment is recurring.

is_cvv_less

is_cvv_less send 0 or 1 for cvv less payment.

installments_number

send the installment number in the request if don't want to show or select installment in card payment page

Response

After a successful request, the server will provide a link to the CURL sender. Merchant website needs to redirect to the site to pay.

The following keys will be available in the response:

KEY	status	string
KEY	success_message	string
KEY	link	string

status

status is the result of the API request. “True” for success and “false” for failed.

success_message

success_message is a string describing status of the request

link

link is the URL where the merchant website needs to redirect for branded 2d/3d payment processing.

Response Redirect

After Payment process, the end user is redirected to the merchant website using parameters given below. For response validation we send a hash_key parameter. Merchant can compare request parameter with hash_key decrypted value that is described in **11.5**

```
$params = [
    'payment_status' => $status,
    'order_no' => $order_id,
    'invoice_id' => invoice_id,
    'status_code' => $error_code,
    'status_description' => $status_message,
    'payment_method' => $payment_type,
    'transaction_type' => $transaction_type,
    'error_code' => $error_code,
    'error' => $status_message,
    'hash_key' => $hash_key,
];
```

payment_status

payment_status = 0/1

0=Failed, 1=Completed

order_id

order_id is Platformode order id.

invoice_id

invoice_id is merchant invoice_id

Response Parameter Explanation

Key	**Explanation**
payment_status	payment_staus can be 1/0. 1=success, 0=fail
order_no	Platformode order no
invoice_id	merchant invoice id
status_code	Platformode Status code, 100 is success code
status_description	Transaction explanation
payment_method	1= Credit Card, 2= Mobile, 3= Wallet
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly. transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	The value of status_code parameter
error	The value of status_description parameter

hash_key	To Validate the request comes from Platformode

Validate Response:

Condition 1 : payment_status == 1 and transaction_type == "Auth" // Transaction is successful and transaction amount is deducted from the card instantly.

Condition 2 : payment_status == 1 and transaction_type == "Pre-Authorization" //Process is successful and transaction amount will be deducted from the card later.

Condition 3 : payment_status == 0 //Transaction is failed

hash_key should be validated on the merchant side to confirm the request comes from Platformode. Also Order Status API(2.1) confirms that the transaction is successful or Failed

2.1 Check Order status

Once payment is completed successfully using 2D/3D, the Platformode payment system redirects users to the merchant's success URL. In the merchant website, order status must be changed to "Completed" and the cart must be cleared. To know the actual order status from Platformode, there should be another request sent to Platformode by invoice id and merchant key. Here is the request and response of the getOrderStatus API:

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/checkstatus	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	invoice_id	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	hash_key	string	Mandatory

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG3M4jfsNeH

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

invoice_id

invoice_id is a unique order id sent by merchant

merchant_key

merchant_key is the unique key of the merchant provided by Platformode.

hash_key

hash_key generation method is given below.

```php
function generateRefundHashKey($invoice_id, $merchant_key, $app_secret) {
    $data = $invoice_id.'|'.$merchant_key;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt(
        "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
    );
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $hash_key = str_replace('/', '__', $msg_encrypted_bundle);
    return $hash_key;
}
```

Response:

Failed Response:

```
{
    "status_code": 41,
    "status_description": "Oder Failed",
    "transaction_status": "Failed",
    "order_id": "162435924998223",
    "transaction_id": "6jvQ7-l3kb-TC10-98950-220621",
    "message": "Oder Failed",
    "reason": "Failed # transaction failed",
    "bank_status_code": "Failed # transaction failed",
    "bank_status_description": "transaction failed",
    "invoice_id": "50781624359247",
    "total_refunded_amount": 0,
    "product_price": "20.00",
    "transaction_amount": 21.2,
    "ref_number": "",
    "transaction_type": "Pre-Authorization",
    "merchant_commission": "2.18"
    "user_commission": "0.96",
    "md_status": 4,
    "original_bank_error_code": "43",
    "original_bank_error_description": "Çalıntı Kart-Karta el koyunuz",



}
```


Success Response:


```
{
    "status_code": 100,
    "status_description": "An order has been taken place for this invoice id: 33491162435928”,
    "transaction_status": "Completed",
    "order_id": "162435932934307",
    "transaction_id": "SXY0m-o3kb-TC10-98950-220621",
    "message": "An order has been taken place for this invoice id: 33491162435928",
    "reason": "",
    "bank_status_code": "",
    "bank_status_description": "",
    "invoice_id": "33491162435928",
    "total_refunded_amount": 0,
    "product_price": "20.00",
    "transaction_amount": 22.1,
    "ref_number": ""
    "transaction_type": "Auth",
    "merchant_commission": "2.68"
    "user_commission": "1.06",
    "md_status": 1,
    "settlement_date": “2022-04-22”,
    "original_bank_error_code": "",
    "original_bank_error_description": "",
```

}

Success Response for Recurring:


```
{
    "status_code": 100,
    "status_description": "An order has been taken place for this invoice id: 16014922137077",
    "transaction_status": "Completed",
    "order_id": "16014922157187",
    "transaction_id": "iH8MK-EQe-TC10-98950-300920",
    "message": "An order has been taken place for this invoice id: 16014922137077",
    "reason": "",
    "bank_status_code": "",
    "bank_status_description": "",
    "invoice_id": "16014922137077",
    "total_refunded_amount": 0,
    "product_price": "100.00",
    "transaction_amount": 100,
    "recurring_id": 303,
    "ref_number": "",
    "recurring_plan_code": "1601492241FdsraX",
    "next_action_date": "2021-05-30 03:10:00",
    "recurring_status": "Active",
    "transaction_type": "Auth",
    "merchant_commission": "7.68",
    "user_commission": "4.06",

}
```


3. Get Token

The API will generate a token to use in the other APIs to validate merchants. It also returns a payment integration option set for the merchant. The response key is "is_3d". The possible values of is_3d are: 0, 1, 2 and 4.

0 = White label 2D only, 1 = White label 2D or 3D , 2 = White label 3D only, 4 = Branded payment solution

If the token api returns 1, the merchant website must display a checkbox for the user to choose 2D or 3D.

Every token has 2 hour expiry time and it is highly recommended to use same token until it is expired. There is no need to call this API everytime for every request to proceed when a token for a merchant is in scope of its validity period.

The following request and response are examples of the API.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/token	*application/json*

Type	Params	Data Type	Condition
KEY	app_id	string	Mandatory
KEY	app_secret	string	Mandatory

app_id

app_id is a unique key assigned to the merchant. It must be sent from a merchant website.

app_secret

app_secret is a unique and secret key given to the merchant. It must be sent from a merchant website.

Response:

Failed Response:

```
{
   "status_code": 2,
   "status_description": "Invalid credentials"
}
```

Success Response:

```
   {
      "status_code": 100,
      "data": {
```

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6IjEwNzNmYjg5NjI2MzlhNzlkZDMxZDlhOWJmMGZiNmI3NGY2YWQ3OTRiYzJlNjRhZDUzMmRlNWUxOTRlZTRhZmY3OTUwZDczOWRjMzVhYzgwIn0.eyJhdWQiOiIxNyIsImp0aSI6IjEwNzNmYjg5NjI2MzlhNzlkZDMxZDlhOWJmMGZiNmI3NGY2YWQ3OTRiYzJlNjRhZDUzMmRlNWUxOTRlZTRhZmY3OTUwZDczOWRjMzVhYzgwIiwiaWF0IjoxNTc0MTQ3OTg0LCJuYmYiOjE1NzQxNDc5ODQsImV4cCI6MTYwNTc3MDM4NCwic3ViIjoiMiIsInNjb3BlcyI6W119.h1HJlmRGI6s3MKIJY-r9fzoBuI82aodvw3cOa10GznSB0eG6vwGSqWpPSLDLKA3dJQgqsX1khAI3L95_nOQEPFSqUcDiQDjNLXE4xE9R7baflTa9LONAo6Tdroq90SainymhVPnpsZnqgvA2vhGAOejyC-CULZ-SwiC1-mUhI0rd-Q2y2Oi1MGlIv32pel5QKQWsfrgCq8ZoruR3Zd_bkYKfdxKEM4OLuyHWcSl6VrqqQBlMbvby2OHCFcTVR3I1KyhAugRG4SG6l8b4ooDmkMjltR3ziBr4sW0l5fwSHcID34UmQ83ut1QHg8U9ZC_GLB6yKO4uXGCI4IoSPqu69f5SfTnNW9n90UVUfC1IRQAMncZLEasoh6goMdCcElmvPIFV-zFskTVbsXJvliG_D91zHyOM4s748HgJizxiPVL1pvk1XRLUSUsulqs1SSP3bDKcfKPT6nRHMFOOUx_ZiLXAFSIgzTz4_liq-GYyHJ_1VKxSftQV0pxBRjm57lBvBn-UIeT7BosYH8EPD2S3KEy8yPlK1lTvo_YogtVkE2t63ZOml1h05r_x8v_CFPhhpSlKAF_a0D_EnJaqrivldZCC7Kgyg_tDojBaDtA4ptmQLiNKs46ZKfEx7ccrwFsXExUIRJ5BiL0c2JqWhP9LTTAsprEPcu5izGnFmZa3OQI",

"is_3d": 1

},

"status_description": "Successfully Generated token"

"expires_at": "2021-01-18 20:20:10"

}

4. Get Installments

The API must be called whenever the user keyin card number and length exceed six digits. getPos API is responsible for providing a list of installments based on the given card number on the checkout page.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/getpos	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	credit_card	digit	Mandatory
KEY	amount	double	Mandatory
KEY	currency_code	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	is_recurring	int	optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be "Bearer"

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2WvzxuxavqSyu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL14FSHyeb3lnuF8VFyERwbf-tAll

`Accept`

Accept Determines what type of representation is desired at client side. The value should be "*application/json*"

credit_card

credit_card determines what type of representation is desired at client side. The value should be "*application/json*".

`Full credit card number must have to send.`

amount

amount is total product amount

currency_code

currency_code is ISO code of the currency. For example, USD, TRY, EUR etc.

merchant_key

merchant_key is the unique key of the merchant provided by Platformode.

is_recurring

is_recurring= 1 is mandatory when payment is recurring.

is_2d

is_2dIn get token API response, if “is_3d” is 0 then is_2d=1 should be sent;

Response:

Failed Response:

```
{
    "status_code": 3,
    "status_description": "merchant not found"
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "Successful",
    "data": [
        {
            "pos_id": 1,
            "campaign_id": 0,
```

"allocation_id": 0,

"installments_number": 1,

"amount_to_be_paid": 23,

"currency_code": "TRY",

"currency_id": 1,

"title": "Single payment",

"hash_key ": '5uUVKijz5im5FfStic2wVX4gG8ngRfMS3H+FvAauQvOc1nAnh9GZ9T6zyxV5WUnQP2F'

}

],

}

Success Response of tarim credit card or bin number

If provided credit card or bin number is Tarim/Agriculture then the success response is

Success Response:

{

"status_code": 100,

"status_description": "Successful",

"data": [

{

"pos_id": 1,

"campaign_id": 0,

"allocation_id": 0,

"installments_number": 1,

"amount_to_be_paid": 23,

"currency_code": "TRY",

"currency_id": 1,

"title": "Single payment",

"hash_key ": '5uUVKijz5im5FfStic2wVX4gG8ngRfMS3H+FvAauQvOc1nAnh9GZ9T6zyxV5WUnQP2F',

"maturity_period": 1,

```
            "payment_frequency": 1
        }
    ],
}
```


Notes:

The data values might be an array. In that case, mechant website should display all the installment options for the user to select. By default, the first installment will be selected. For any case, there will be at least one installment available in the response.

Based on client request we added card_issuer_name in response for specific brand.

7.0 Plugins

The following plugins are available to install on merchant website without writing custom code to integrate Platformode payment system:

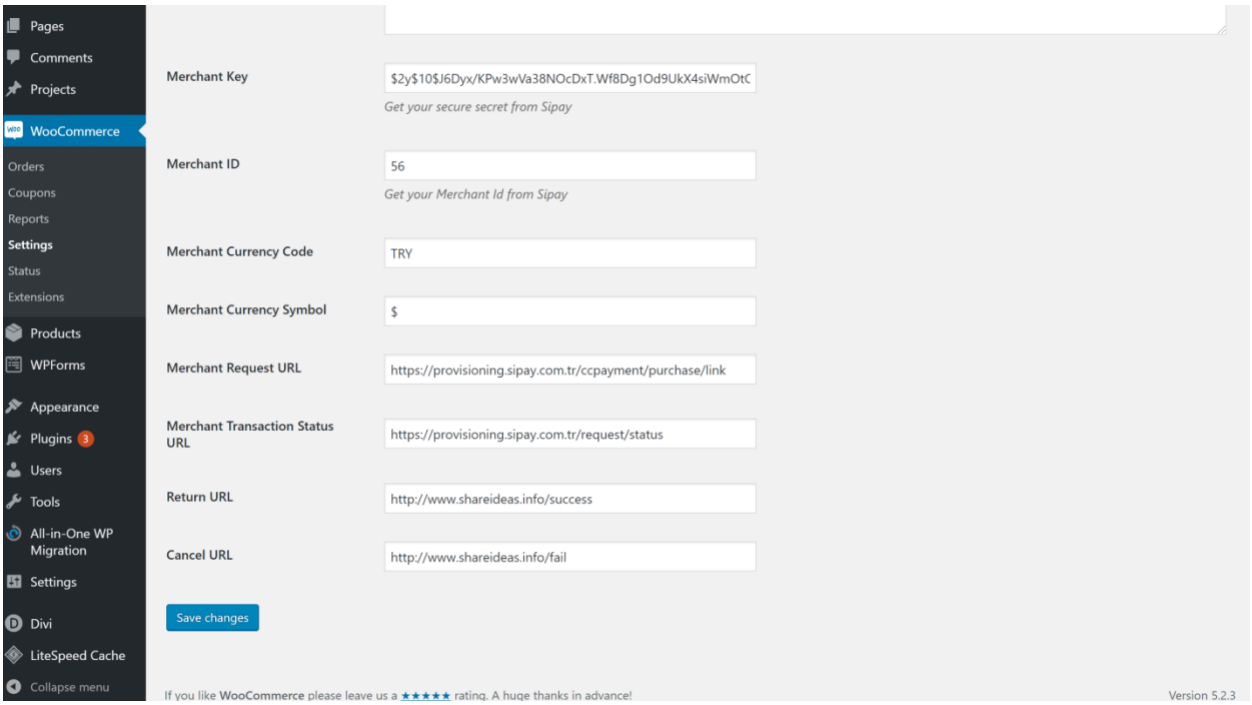
3.1 WooCommerce plugin

Follow the steps below to install the WooCommerce plugin:

Step#1: Go to plugins=>add plugin

Step#2: browse the plugin zip file and activate.

Step#3: After successful activation, the following screen should be appeared to configure the plugin



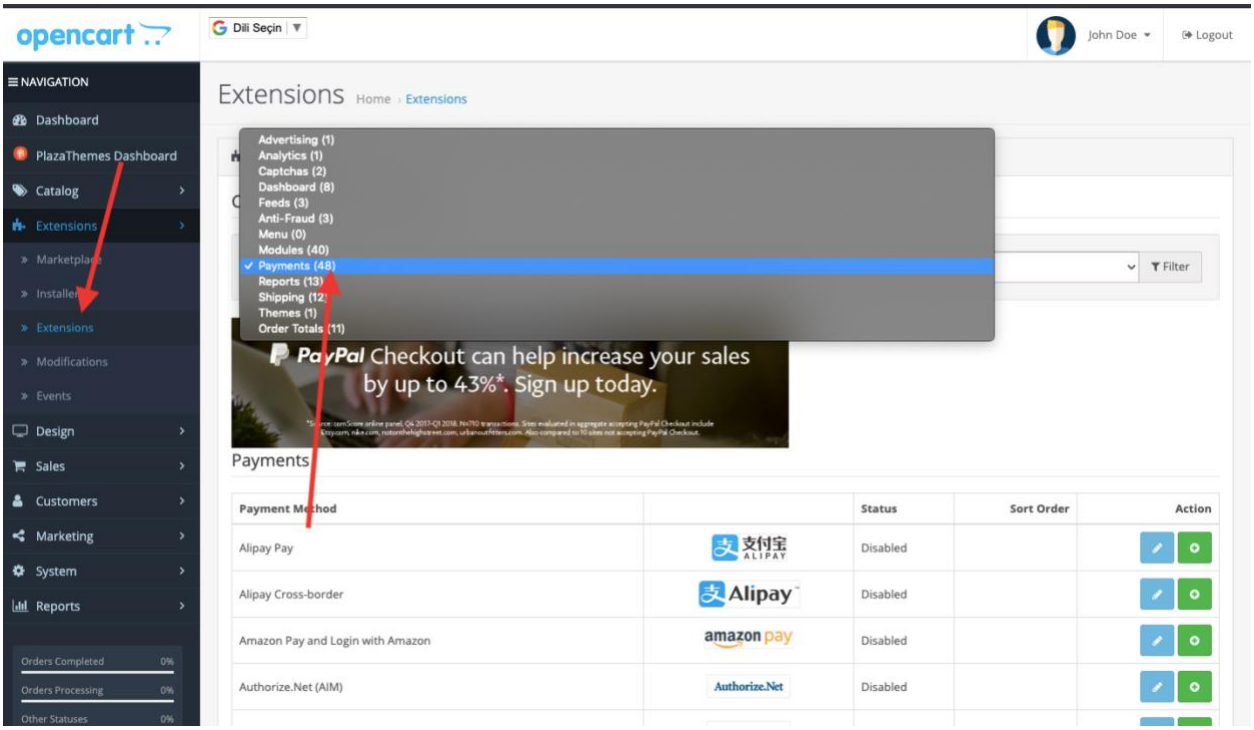
Step#4: Configure the key, return url, cancel url etc. and click the "Save Changes". On the checkout page, Platformode payment option should be appeared like below:

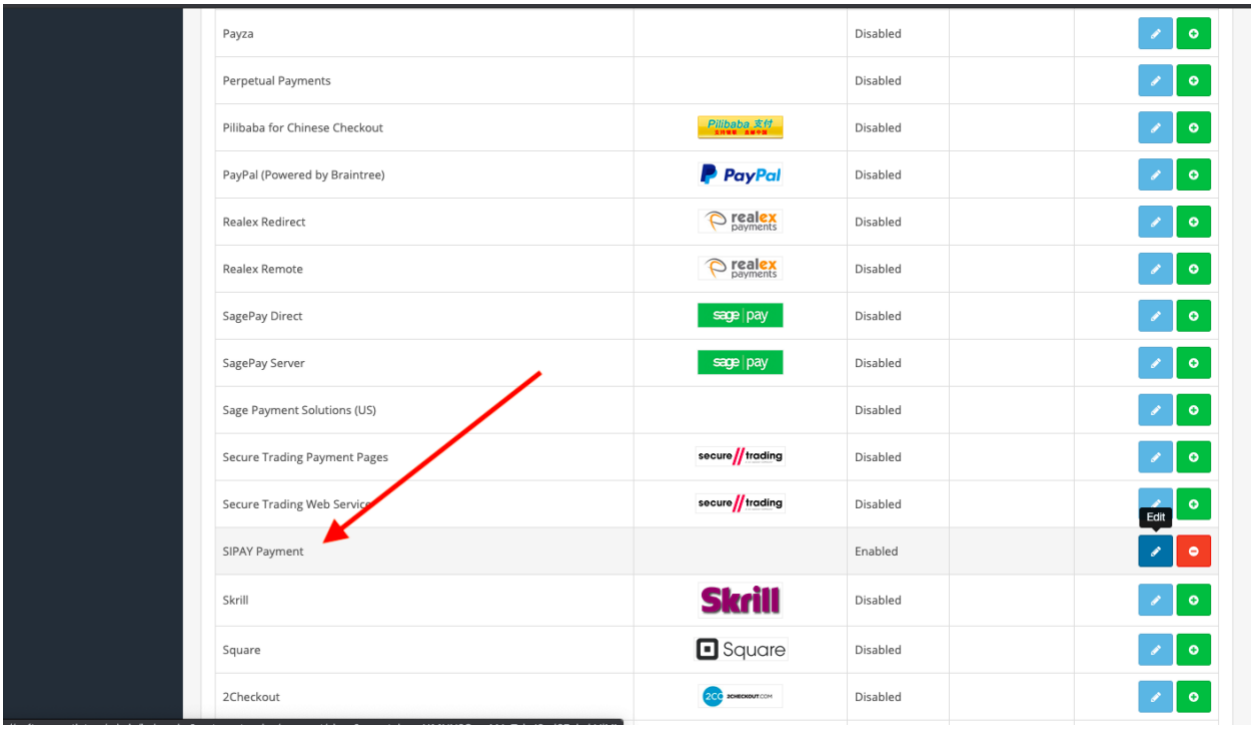
3.2 Opencart module

1. Upload zip file to your Host.

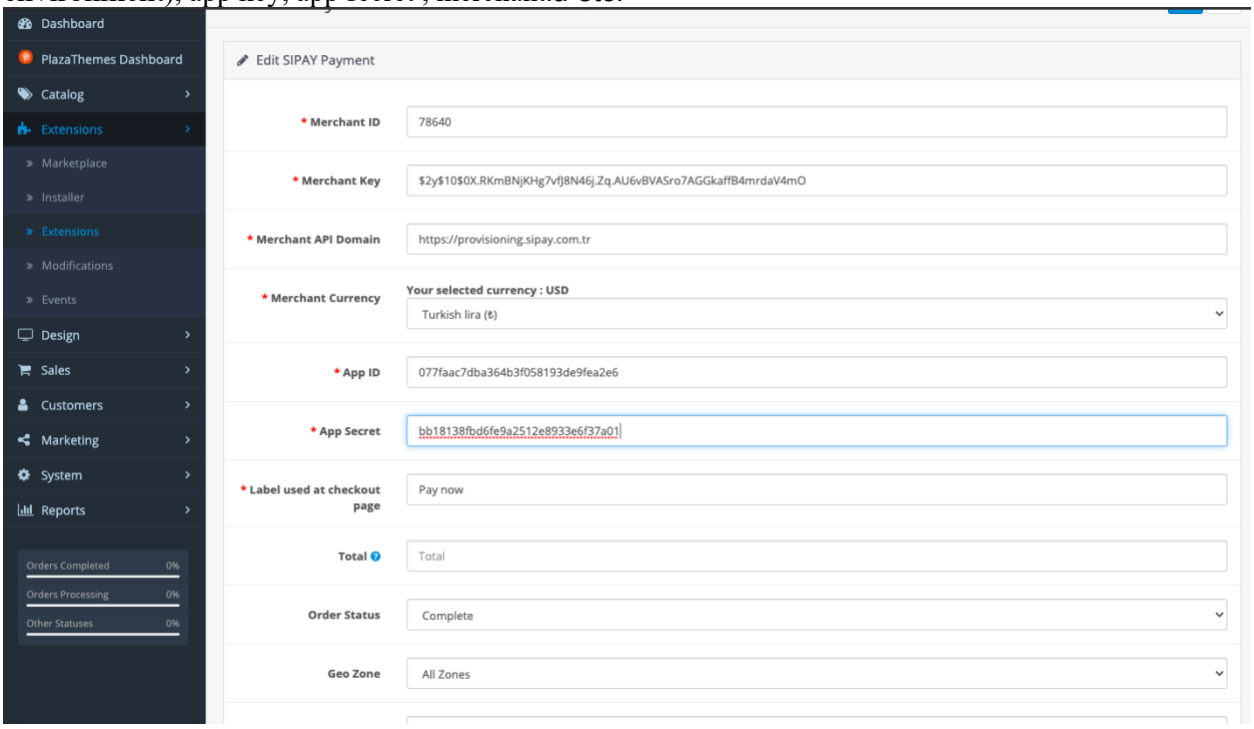
2. Extract zip file after that logged in your OpenCart 3.x admin

3.Go to extension => select payment and search Platformode and press + button to install.



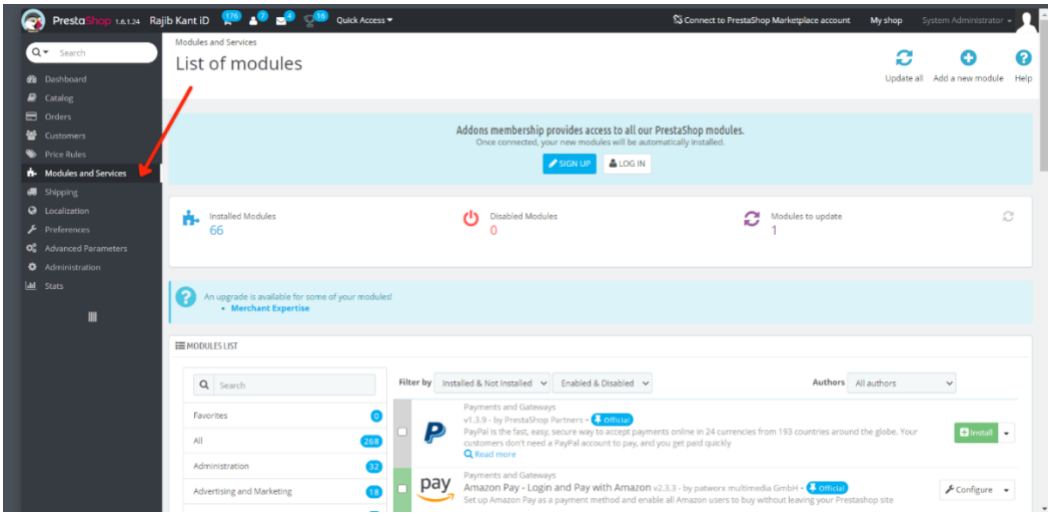


4. When installed you will see an edit button to do settings. Click on that and set api domain (https://app.Platformode.com.tr for production and http://fpprov.softrobotics.lt for test environment), app key, app secret , merchantid etc.

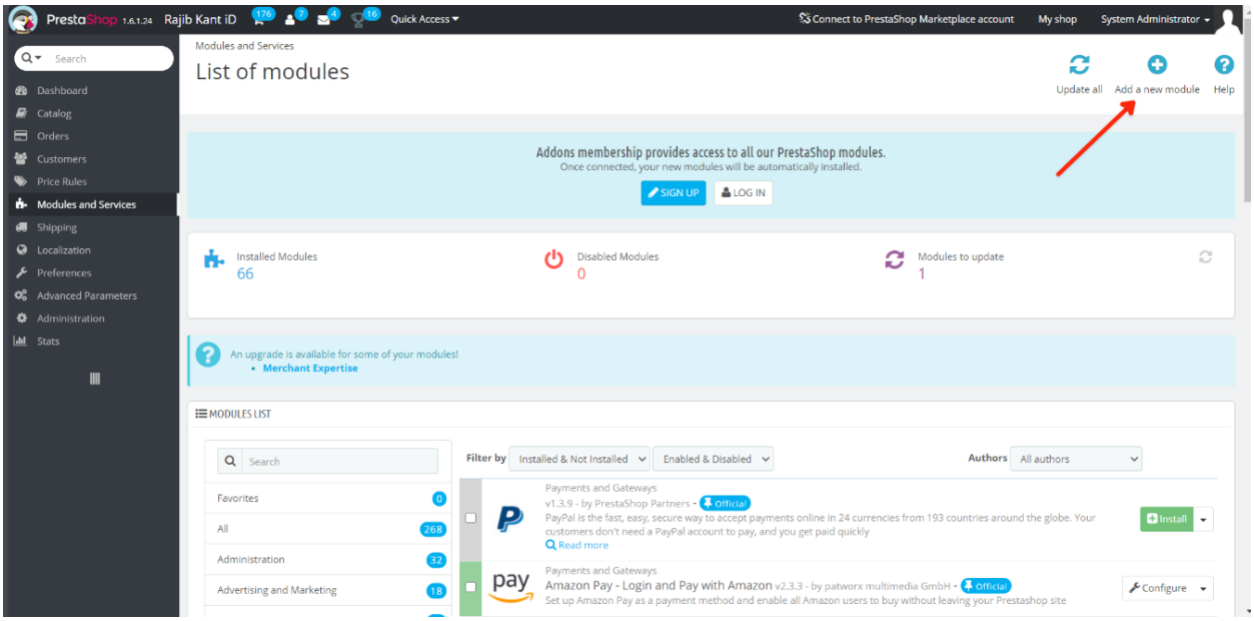


3.3 Presta Module

- Go to Presta Shop's admin page.
- Click on Modules and Services> Modules and Services



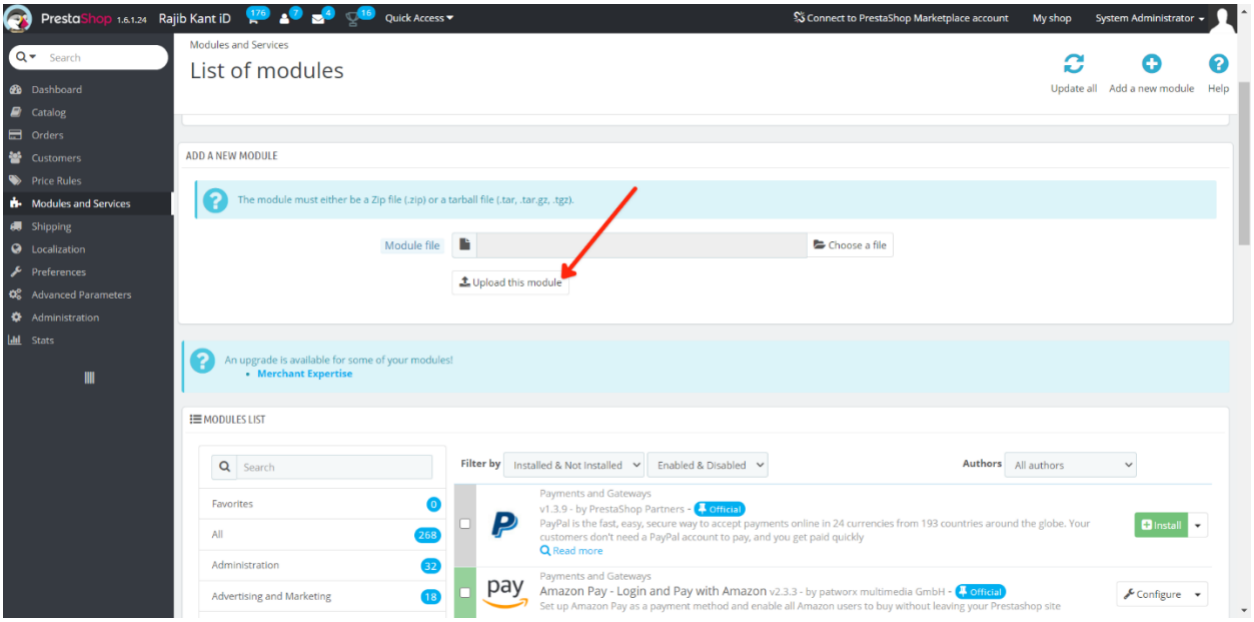
- Click on the "**Add a new module** " button

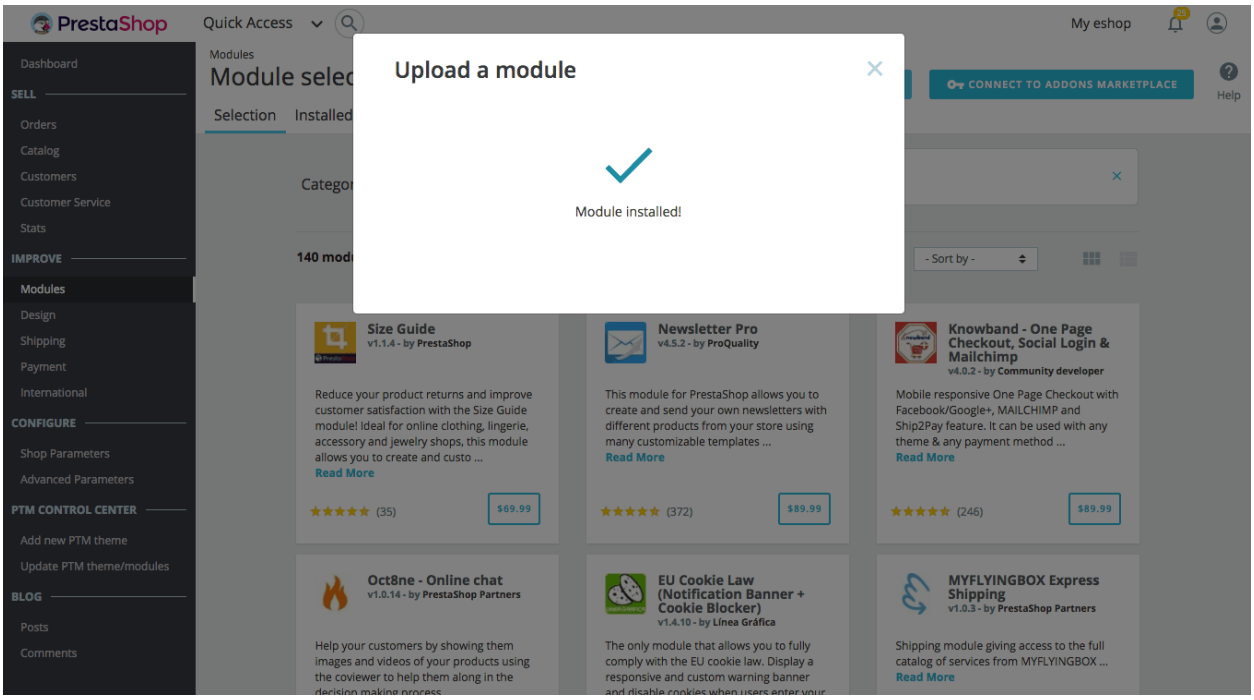


Click the "choose a file" button

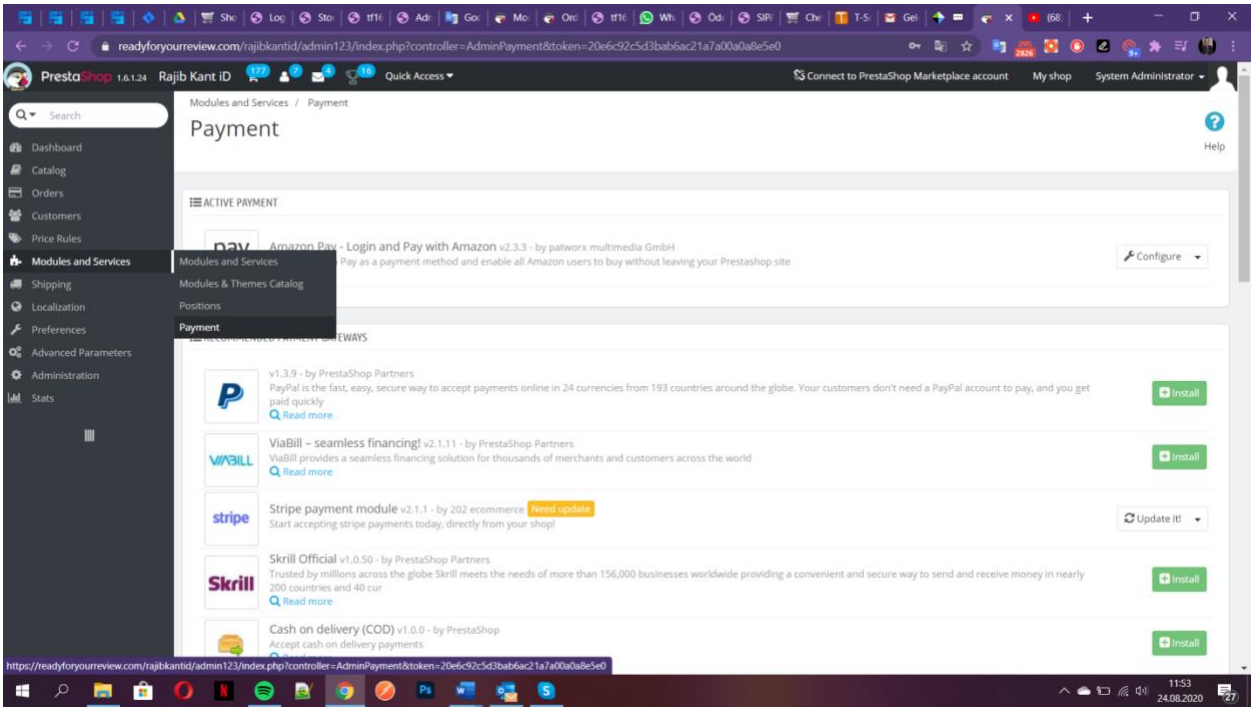
- Add the file of the Platformode module.

Then click on "Upload this module".

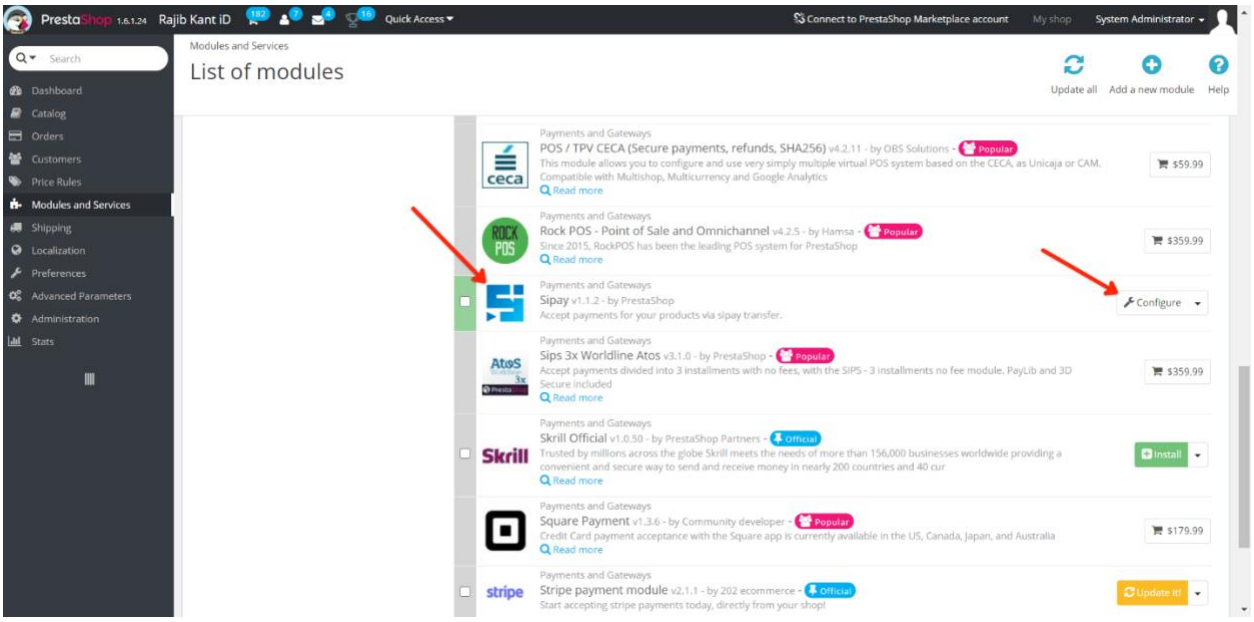




- Click on Modules and Services> Payment to make the settings when it is installed.

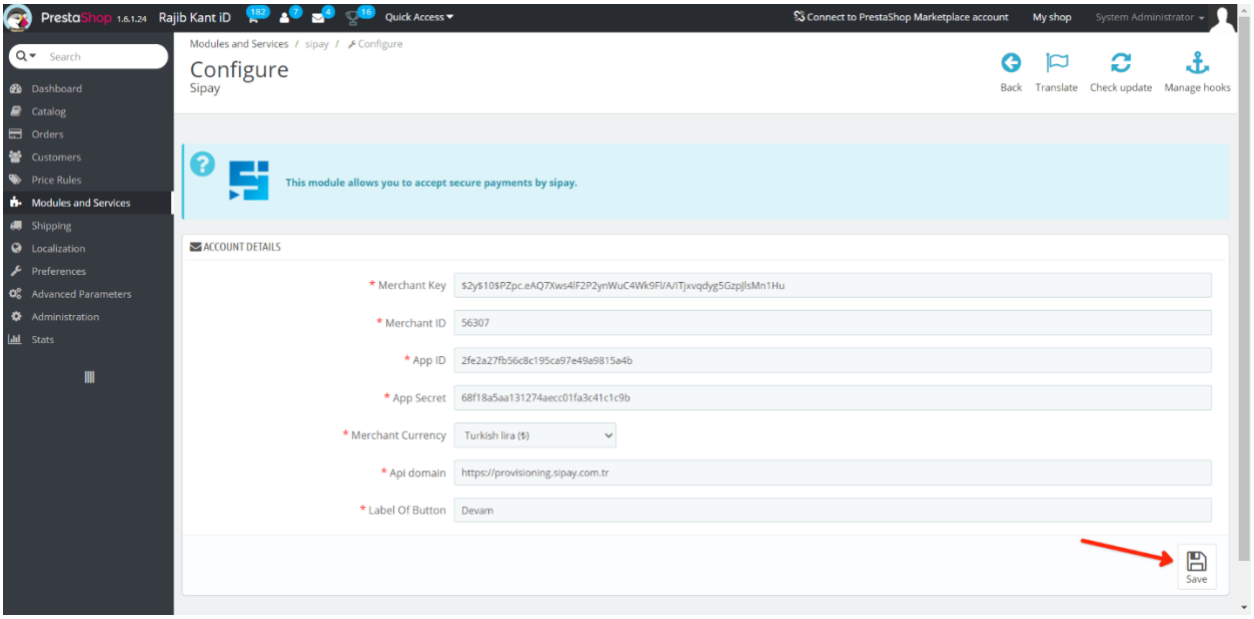


- Click on the configuration next to the order module.



Click on that and set api domain (https://app.Platformode.com.tr for production and http://testapp.platformode.com.tr for test environment), app key, app secret , merchant id etc.

It is saved with the save button below.



10.0. Transaction Refund

Method	URL	Content-Type
POST	<ACCESS_URL>/api/refund	*Application/json*

Type	Params	Data Type	Condition

HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	invoice_id	string	Mandatory
KEY	amount	double/empty string	Mandatory
KEY	app_id	string	Mandatory
KEY	app_secret	string	Mandatory
KEY	merchant_key	string	Mandatory
Key	hash_key	string	Mandatory
Key	refund_transaction_id	string	Mandatory
key	refund_web_hook_key	string	Optional

Authorization

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2WvzxuxavqSyu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL14FSHyeb3lnuF8VFyERwbf-tAll

Accept

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

invoice_id

invoice_id is a unique order id sent by merchant

amount

amount to be refunded

NB : If amount is set to empty string ("") the rest of the amount of a transaction will be refunded.
Here, rest of the amount = total amount of a transaction – total amount of partial refunds

app_id

app_id is the unique id provided by Platformode

app_secret

app_secret is a secret key provided by Platformode .

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

Hash Key Generation Method

```
function generateRefundHashKey($amount, $invoice_id, $merchant_key,
$app_secret) {

    $data = $amount.'|'.$invoice_id.'|'.$merchant_key;

    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);

    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);

    $encrypted = openssl_encrypt(
        "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
    );

    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $hash_key = str_replace('/', '__', $msg_encrypted_bundle);
```

```
    return $hash_key;
}
```

Response:

Failed Response:

```
{
    "status_code": 49,
    "status_description": "Refund Failed",
    "order_no": "15925741639038",
    "invoice_id": "66955",
    "ref_no": ""
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "Refund completed successfully",
    "order_no": "15925741639038",
    "invoice_id": "66955",
    "ref_no": "5454545dgdgd545545"
}
```

Refund WebHook

The Merchant needs to set up your refund web hook URL(key, value) in the Platformode merchant panel at **https://app.Platformode.com.tr/merchant/apisetting** . For refund, *refund_web_hook_key* key should be sent with the refund request parameter. Platformode validates that the key exists in the database while making a refund request.

At refund approval, it is sent a POST request to a merchant refund web hook url with following parameters given below.

Type	Parameters	Sample Value
Key	invoice_id	8iu75g
Key	order_id	15767887576675

Key	amount	10.50
Key	status	Completed
Key	hash_key	3rter3dtrytyutuer5y5tyttyrtryryry=

Refund WebHook Response Validation Using Hash key.

In Refund Approval, Platformode sends a post request to the web hook url . The problem with these links is that they can be accessed by an anonymous person. To prevent this problem, it is recommended to validate the request using hash key, since it is added some parameters to the links such as status, invoice_id, order_id, amount and hash_key with the request parameters.

```
function validateHashKey($hash_key, $secret_key){
        $status = '';
        $amount = $invoice_id = $order_id = 0;
        if (!empty($hash_key)) {
        $hash_key = str_replace('__', '/', $hash_key);
        $password = sha1($secret_key);
        $components = explode(':', $hash_key);
        if (count($components) >2) {
        $iv = $components[0] ?? '';
        $salt = $components[1] ?? '';
        $salt = hash('sha256', $password . $salt);
       $encrypted_msg = $components[2] ?? '';
        $decrypted_msg = openssl_decrypt($encrypted_msg, 'aes-256-cbc', $salt, null, $iv);
        if (strpos($decrypted_msg, '|') !== false){
```

```
        $array = explode('|', $decrypted_msg);
        $status = $array[0] ?? 0;
        $amount = $array[1] ?? 0;
        $invoice_id = $array[2] ?? '0';
        $order_id = $array[3] ?? 0;
        }
        }
        }
        return [$status, $amount, $invoice_id, $order_id];
}
```

Here, $hash_key must be taken from a request and $secret_key is a merchant app secret that was provided from Platformode.

11.0. Recurring WebHook

First of all, you need to set up your recurring web hook URL(key, value) in the Platformode merchant panel at **https://app.Platformode.com.tr/merchant/apisetting** . For recurring payment, merchants must set *recurring_web_hook_key*with invoice. Platformode validates that this key exists in the database while making a purchase request.

For each recurring payment, SPlatformode sends a POST request to a merchant recurring web hook url with following parameters given below.

Type	Parameters	Sample Value
Key	merchant_key	$2hiu3er3iejdfjkvi343hh553h34k34h
Key	invoice_id	8iu75g
Key	order_id	15767887576675

Key	product_price	10.50
Key	plan_code	15767Ythgyuy
Key	recurring_number	2
Key	status	“Completed”/ “Failed”
Key	attempts	10
Key	action_date	20-08-19 14:15:48
Key	hash_key	Ssfs:sdfsfs:4rfddgddfdgdgdg=

Suggested guideline to implement web hook:

Step 1: validate request is POST and merchant key is valid.

Step 2: After that, call recurring query API.

Method	URL	Content-Type
POST	<ACCESS_URL>/api/recurringPlan/query	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	plan_code	string	Mandatory
KEY	recurring_number	integer	Mandatory

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

plan_code

plan_code is a unique key that was generated during the first payment.(In this case, plan_code should be taken from web hook response).

recurring_number

recurring_number must be taken from a webHook request sent by Platformode at every recurrence.

Response Example:

Response body


```
{
  "status_code": 100,
  "message": "Recurring created successfully",
  "recurring_id": 31,
  "plan_code": "15978357430kZFft",
  "currency": "TRY",
  "currency_symbol": "₺",
  "first_amount": 20,
  "recurring_amount": 20,
  "total_amount": 120,
  "payment_number": 6,
  "payment_interval": 1,
  "payment_cycle": "DAY",
  "first_order_id": "15978356611194",
  "merchant_id": 56307,
  "card_no": "5406****5403",
  "next_action_date": "2020-09-04",
  "recurring_status": "Active",
  "transaction_date": "2020-08-19 14:15:45",
  "transactionHistories": [
    {
      "id": 99,
      "sale_recurring_id": 31,
      "sale_id": 16628,
      "merchant_id": 56307,
      "action_date": "2020-08-19 14:15:45",
      "status": "Success",
      "recurring_number": 1
    }
  ]
}
```


Step 3: Create transactions on your site according to “recurring_number” and “status”.

11.0.1 Update Recurring

Recurring payment can be updated using this API.

Method	URL	Content-Type
POST	<ACCESS_URL>/api/recurringPlan/update	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	plan_code	string	Mandatory
KEY	recurring_payment_number	integer	Mandatory
KEY	recurring_status	string	Optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be "Bearer"

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

`Accept`

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

plan_code

plan_code is a unique key that was generated during the first payment.(In this case, plan_code should be taken from web hook response).

recurring_payment_number

recurring_number must be taken from a webHook request sent by Platformode at every recurrence.

recurring_status

recurring_status param value is active/inactive.

Request Sample :

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he",
  "plan_code": "1649229007oFOkJ8",
  "recurring_amount": "100.00",
  "recurring_status": "ACTIVE",
  "recurring_payment_number": "2"
}
```

Response:

Failed Response:

```
{
 "status_code": 71,
 "message": "Recurring plan update failed"
}
```

Success Response:

```
{
 "status_code": 100,
 "message": "Recurring Updated successfully"
}
```

11.0.2 Query Recurring Information

A particular recurring payment information can be fetched by this API.

Method	URL	Content-Type
POST	<ACCESS_URL>/api/recurringPlan/query	*application/json*

Type	Params	Data Type	Condition

HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	plan_code	string	Mandatory

Authorization

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

Accept

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

plan_code

plan_code is a unique key that was generated during the first payment.(In this case, plan_code should be taken from web hook response).

Request Sample :

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he",
  "plan_code": "1649229007oFOkJ8",
}
```

Response:

Failed Response:


```
{
  "status_code": 70,
  "message": "Recurring Not Found",
  "plan_code": "1649229007oFOkJ8df"
}
```


Success Response:


```
{
  "status_code": 100,
  "message": "Recurring card Found",
  "recurring_id": 328,
  "plan_code": "1649229007oFOkJ8",
  "currency": "TRY",
  "first_amount": 103,
  "recurring_amount": 100,
  "total_amount": 503,
  "payment_number": 5,
  "payment_interval": 1,
  "payment_cycle": "MONTH",
  "merchant_id": 98950,
  "recurring_status": "Active",
  "transaction_date": "2022-04-06T07:10:07.000000Z",
  "recurringCardList": [
    {
      "sale_recurring_id": 328,
      "card_token": "f02102a7-6d5e-4200-b259-5fcbcad06595",
      "card_user_key": "195f8bfb-9632-4ef5-aca6-2a9e441c878c",
      "status": "Active",
      "created_at": "2022-04-06T07:10:07.000000Z",
      "updated_at": "2022-04-06T07:10:07.000000Z"
    }
  ]
}
```


11.0.3 Delete Recurring

A recurring payment can be deleted using this API .

Method	URL	Content-Type
POST	<ACCESS_URL>/api/recurringPlan/delete	*application/json*

Type	Params	Data Type	Condition

HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	plan_code	string	Mandatory

Authorization

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

Accept

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

plan_code

plan_code is a unique key that was generated during the first payment.(In this case, plan_code should be taken from web hook response).

Request Sample :

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he",
  "plan_code": "1649229007oFOkJ8",
}
```

Response:

Failed Response:


```
{
  "status_code": 72,
  "message": "Failed to delete"
}
```


Success Response:


```
{
  "status_code": 100,
  "message": "Recurring plan Deleted successfully"
}
```


11.1. Sale WebHook

First of all, you need to set up your sales web hook URL(key, value) in the Platformode merchant panel at **http://app.Platformode.com.tr/merchant/apisetting** . To get this feature, merchants need to send *sale_web_hook_key*with invoice while sending purchase request. This key is optional but if sent, it must be a valid key.

For each payment, we send a POST request to your sale web hook url with following parameters given below.

Headers:

'Content-Type: application/x-www-form-urlencoded'

Type	Parameters	Sample Value
Key	invoice_id	8iu75g
Key	order_id	15767887576675
Key	status	“Completed”/ “Failed”
KEY	payment_status	1=Completed, 0 = Failed

KEY	status_description	Transaction Failed due to ……
KEY	payment_method	1= Card, 2=Moblie, 3=Wallet
KEY	Hash_key	Gfh:rtete:tertetetetrtyyfghhgjjjggtgh=
KEY	transaction_type	“Auth” / ”Pre-Authorization”

Response Parameter Explanation

Key	**Explanation**
payment_status	payment_staus can be 1/0. 1=success, 0=fail
order_no	Platformode order no
invoice_id	merchant invoice id
status_code	Platformode Status code, 100 is success code
status_description	Transaction explanation
payment_method	1= Credit Card, 2= Mobile, 3= Wallet
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly. transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	The value of status_code parameter
error	The value of status_description parameter
hash_key	To Validate the request comes from Platformode
merchant_commission	The merchant commission from the sale transaction
user_commission	The user commssion from the sale transaction

md_status	md_status from the sale transaction

Validate Response:

Condition 1 : payment_status == 1 and transaction_type == “Auth” // Transaction is successful and transaction amount is deducted from the card instantly.

Condition 2 : payment_status == 1 and transaction_type == “Pre-Authorization” //Process is successful and transaction amount will be deducted from the card later.

Condition 3 : payment_status == 0 //Transaction is failed

hash_key should be validated on the merchant side to confirm the request comes from Platformode. Also Order Status API(2.1) confirms that the transaction is successful or Failed

11.2. Merchant Active Installments

Method	URL	Content-Type
POST	<ACCESS_URL>/api/installments	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	merchant_key	string	Mandatory

Authorization

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU

xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

`Accept`

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

merchant_key

merchant_keyis unique key of the merchant provided by Platformode .

Response:

Failed Response:

```
{
    "status_code": 62,
    "message": "Merchant was not assigned any installment yet"
}
```

Success Response:

```
{
    "status_code": 100,
    "message": "Merchant Active Installment",
    "installments": [
        1,
        2,
        3
    ]
}
```

11.3. Pay With White Label Smart 3D

The paySmart3D API is used to submit order and credit card details information to Platformode payment integration system. The user will be redirected to the bank page after the merchant website submits a payment form. Payment would be verified by a SMS code at the bank gateway. Once payment is successful, the user will be

redirected back to the merchant success URL, otherwise, it will be redirected to cancel the URL set by the merchant. In this payment API, there is no need to call getPos Api like other payment api calls.

Special Notes:

1. **Do not use ajax requests to call paySmart3D api. It must be normal form submission to "<ACCESS_URL>/api/paySmart3D"**
2. **Follow sample code given as example file**

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/paySmart3D	*application/x-www-form-urlencoded*

Type	Params	Data Type	Condition
KEY	cc_holder_name	string|max:100	Mandatory
KEY	cc_no	string	Mandatory
KEY	expiry_month	string	Mandatory
KEY	expiry_year	string	Mandatory
KEY	cvv	string	Mandatory
KEY	currency_id	string	Mandatory
KEY	currency_code	string	Mandatory
KEY	installments_number	number	Mandatory
KEY	invoice_id	string	Mandatory
KEY	invoice_description	string	Mandatory
KEY	name	string	Mandatory
KEY	surname	string	Madatory
KEY	total	double	Mandatory
KEY	merchant_key	string	Mandatory

KEY	items	string	Mandatory
KEY	cancel_url	string	Mandatory
KEY	return_url	string	Mandatory
KEY	hash_key	string	Mandatory
KEY	bill_address1	string	Optional
KEY	bill_address2	string	Optional
KEY	bill_city	string	Optional
KEY	bill_postcode	string	Optional
KEY	bill_state	string	Optional
KEY	bill_country	string	Optional
KEY	bill_email	string	Optional
KEY	bill_phone	string	Optional
KEY	sale_web_hook_key	string	Optional
KEY	card_program	string	Optional
KEY	ip	string	Optional
KEY	transaction_type	string	Optional
KEY	payment_completed_by	string	Optional

Request for Recurring

KEY	order_type	*Integer*	Mandatory
KEY	recurring_payment_number	*Integer*	Mandatory
KEY	recurring_payment_cycle	*string*	Mandatory
KEY	recurring_payment_interval	integer	Mandatory
KEY	recurring_web_hook_key	string	Mandatory

Request for Agriculture/Tarim

KEY	*maturity_period*	*Integer*	*Mandatory*
KEY	*payment_frequency*	*Integer*	*Mandatory*
KEY	*response_method*	*String: POST/GET*	*Optional*

Notes:

cancel_url is used to redirect when payment is failed and return_url is used to redirect when payment is successful.

Regardless of fail or success payment, the following keys are available in the query string with cancel and success url:

payment_status, order_id and invoice_id. For example, if the successs URL is https://<my-domain.com/, then after successful payment, Platformode will redirect to https://<my-domain.com/?payment_status=1&order_no=234234232&invoice_id=73434

In the query string, if the payment_status = 1, the getOrderStatus API must be called to clear cart and change order status to "Completed"

name

name First name of the person. For example, if the name of the person who is buying the product is "john Dao", then name should be "john"

surname

surname Last name of the person. For example, if the name of the person who is buying the product is "john Dao", then the surname should be "Dao".

sale_web_hook_key

*sale_web_hook_key*is an optional key. When a purchase request is completed, Platformode sends a post request. So that merchant can perform an event on their site. Platformode validates that this key must exist in the database. Merchant must assign the Sale web hook URL on the Platformode Merchant Panel against this key.

order_type

If *order_type=1*, Platformode validates payment for recurring. Then *recurring_payment_number, recurring_payment_cycle, recurring_payment_interval*keys should not be empty.

Card_program value must be one of following: "WORLD", "BONUS", "MAXIMUM",

"BANKKART_COMBO", "PARAF", "AXESS", "ADVANT", "CARD_FNS"

transaction_type

For *transaction_type* “PreAuth”, a pending transaction is created in the system. But later it is converted to Completed by the merchant's confirmation.

payment_completed_by

Two values are applicable “*merchant*” or “*app*”

For “*payment_completed_by*” => “*merchant*”, payment acquisition/capture/complete stage has to be performed by merchant via CompletePayment API. In case of “*app*” (which is also default by value) the payment will be completed via PFMS/Application. If merchant sends “merchant” and doesn’t complete the payment via API, the transaction will be automatically converted to fail payment after 15 minutes of its staging time.

maturity_period

An integer number have to send if transaction is agriculture or tarim. maturity_period

Values can be obtained by calling get installment api by using tarim card or bin number.

payment_frequency

An integer number have to send if transaction is agriculture or tarim. payment_frequency

Values can be obtained by calling get installment api by using tarim card or bin number.

recurring_payment_number

*recurring_payment_number*defines installment count. If first_amount is $100 and **recurring_payment_number**is 5, then the total amount will be deducted as $100*5 = $500. (Cost of transaction may be added with each transaction)

recurring_payment_cycle

recurring_payment_cycle defines the unit type of *recurring_payment_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**: Years

recurring_payment_interval

*recurring_payment_interval*defines interval value. If *recurring_payment_interval*= 2 and *recurring_payment_cycle* = “M” then transaction will occur once in every 2 months.

recurring_web_hook_key

*recurring_web_hook_key*defines merchant recurring web hook url . An URL must be assigned on the Platformode Merchant Panel against this key. Platformode validates this key must exist in the database and it is a required value when payment is recurring.

hash_key

*hash_key*is declared to secure the payment. End user may change the product price before going to the bank. Here is the algorithm to write the hash key given below.

```
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret){

$data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;

$iv = substr(sha1(mt_rand()), 0, 16);
$password = sha1($app_secret);

$salt = substr(sha1(mt_rand()), 0, 4);
$saltWithPassword = hash('sha256', $password . $salt);

$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);

$msg_encrypted_bundle = "$iv:$salt:$encrypted";
$msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

return $msg_encrypted_bundle;
}
```

response_method
Based on the value (GET/POST) of this parameter response will be sent as http GET or POST. By default, it is GET.

Response Redirect

After Payment process, the end user is redirected to the merchant website using parameters given below. For response validation we send a hash_key parameter. Merchant can compare request parameter with hash_key decrypted value that is described in **11.5**

```
$params = [
    'payment_status' => $status,
    'order_no' => $order_id,
    'invoice_id' => invoice_id,
    'status_code' => $error_code,
    'status_description' => $status_message,
    'payment_method' => $payment_type,
    'transaction_type' => $transaction_type,
    'error_code' => $error_code,
    'error' => $status_message,
    'hash_key' => $hash_key,

];
```

payment_status

payment_status = 0/1

0=Failed, 1=Completed

order_id

order_id is Platformode order id.

invoice_id

invoice_id is merchant invoice_id

Response Parameter Explanation

Key	**Explanation**
payment_status	payment_staus can be 1/0. 1=success, 0=fail
order_no	Platformode order no
invoice_id	merchant invoice id
status_code	Platformode Status code, 100 is success code
status_description	Transaction explanation
payment_method	1= Credit Card, 2= Mobile, 3= Wallet
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly. transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	The value of status_code parameter
error	The value of status_description parameter
md_status	Card authentication status
hash_key	To Validate the request comes from Platformode

Validate Response:

payment_status == 1 and transaction_type == “Auth” // Transaction is successful and transaction amount is deducted from the card instantly.

payment_status == 1 and transaction_type == “Pre-Authorization” //Process is successful and transaction amount will be deducted from the card later.

payment_status == 0 //Transaction is failed

hash_key should be validated on the merchant side to confirm the request comes from Platformode. Also Order Status API(2.1) confirms that the transaction is successful or Failed

11.4. Merchant Commission API

Merchant commission API returns merchant commission and end user commission by card program.

Method	URL	Content-Type
GET	<ACCESS_URL>/api/commissions	*application/json*

Type	Parameters	Sample Value
Key	currency_code	TRY, USD, EUR

Authorization

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.

Accept

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

currency_code

By default, it is set as “TRY”. Platformode Only allows “TRY”, “USD”, “EUR”

Response:

Failed Response:


```
{
    "status_code": 0,
    "status_description": "No Data Found"
}
```


Success Response:{


```
    "status_code": 100,
    "status_description": "successful",
    "data": {
        "1": [
            {
                "card_program": "BANKKART_COMBO",
                "merchant_commission_percentage": "5.0000",
                "merchant_commission_fixed": "1.0000",
                "user_commission_percentage": 3,
                "user_commission_fixed": 1,
                "currency_code": "TRY",
                "installment": 1
            },
            {
                "card_program": "BONUS",
                "merchant_commission_percentage": "0.0000",
                "merchant_commission_fixed": "0.0000",
                "user_commission_percentage": 0,
                "user_commission_fixed": 0,
                "currency_code": "TRY",
                "installment": 1
            }
        ],
}
```


N.B : if commission comes `"x"` for a card program under an installment, it means the card program is not enabled for the installment.

11.5. Response Validation Using Hash key.

In 3d payment, after payment completion, it is redirected to merchant success or failure link. The problem with these links is that they can be accessed by an anonymous person. To prevent this problem, it is recommended to validate the request using hash key, since it is added some parameters to the links such as status, invoice_id, order_id and hash_key while redirecting from Platformode.

Moreover, **in recurring payment,** for every recurrence, it is sent a post request to the merchant webhook. As usual, a webhook is an open link. So, recurrence requests can be validated using hash_key as well. The validation process sample code is given below.

```
function validateHashKey($hash_key, $secret_key){

$status = $currency_code = '';
$total = $invoice_id = $order_id = 0;

   if (!empty($hash_key)) {
$hash_key = str_replace('__', '/', $hash_key);
$password = sha1($secret_key);

$components = explode(':', $hash_key);
      if (count($components) >2) {
$iv = $components[0] ?? '';
$salt = $components[1] ?? '';
$salt = hash('sha256', $password . $salt);
$encrypted_msg = $components[2] ?? '';

$decrypted_msg = openssl_decrypt($encrypted_msg, 'aes-256-cbc', $salt, null, $iv);

         if (strpos($decrypted_msg, '|') !== false){
$array = explode('|', $decrypted_msg);
$status = $array[0] ?? 0;
$total = $array[1] ?? 0;
$invoice_id = $array[2] ?? '0';
$order_id = $array[3] ?? 0;
$currency_code = $array[4] ?? '';
}
      }
   }
```

```
return [$status, $total, $invoice_id, $order_id, $currency_code];
}
```

Here, $hash_key must be taken from a request and $secret_key is a merchant app secret that was provided from Platformode.

N.B: $status = 0(Failed) or $status = 1(Completed)for 3d payment. But for recurring webhook and sale webhook, $status = Completed

or $status = Failed

11.6. Pay With White Label Smart 2D

The pay API is used to submit order and credit card details information to the Platformode payment integration system. Merchant website should receive payment status immediately without loading the checkout page. Based on API success status, cart and order status must be changed accordingly. In this payment API, there is no need to call getPos Api like other payment api calls.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/paySmart2D	*application/json*

Type	Params	Data Type	Condition
KEY	cc_holder_name	string|max:100	Mandatory
KEY	cc_no	string	Mandatory
KEY	expiry_month	string	Mandatory
KEY	expiry_year	string	Mandatory
KEY	cvv	string	Mandatory
KEY	currency_id	string	Mandatory
KEY	currency_code	string	Mandatory
KEY	installments_number	number	Mandatory

KEY	invoice_id	string	Mandatory
KEY	invoice_description	string	Mandatory
KEY	name	string	Mandatory
KEY	surname	string	Mandatory
KEY	total	double	Mandatory
KEY	merchant_key	string	Mandatory
KEY	items	string	Mandatory
KEY	cancel_url	string	Optional
KEY	return_url	string	Optional
KEY	hash_key	string	Mandatory
KEY	bill_address1	string	Optional
KEY	bill_address2	string	Optional
KEY	bill_city	string	Optional
KEY	bill_postcode	string	Optional
KEY	bill_state	string	Optional
KEY	bill_country	string	Optional
KEY	bill_email	string	Optional
KEY	bill_phone	string	Optional
KEY	sale_web_hook_key	string	Optional
KEY	card_program	string	Optional
KEY	ip	string	Optional
KEY	transaction_type	string	Optional
KEY	vpos_type	string	Optional
KEY	identity_number	string	Optional

Request for Recurring

KEY	order_type	*Integer*	Mandatory
KEY	recurring_payment_number	*Integer*	Mandatory
KEY	recurring_payment_cycle	*string*	Mandatory
KEY	recurring_payment_interval	integer	Mandatory
KEY	recurring_web_hook_key	string	Mandatory

Notes:

`name`

name First name of the person. For example, if the name of the person who is buying the product is “john Dao”, then name should be “john”

`surname`

surname Last name of the person. For example, if the name of the person who is buying the product is “john Dao”, then the surname should be “Dao”.

sale_web_hook_key

*sale_web_hook_key*is an optional key. When a purchase request is completed, Platformodesends a post request. So that merchant can perform an event on their site. Platformodevalidates that this key must exist in the database. Merchant must assign the Sale web hook URL on the PlatformodeMerchant Panel against this key.

order_type

If *order_type=1*, Platformodevalidates payment for recurring. Then *recurring_payment_number, recurring_payment_cycle, recurring_payment_interval*keys should not be empty.

card_program value must be one of following: "WORLD", "BONUS", "MAXIMUM", "BANKKART_COMBO", "PARAF", "AXESS", "ADVANT", "CARD_FNS"

transaction_type

For *transaction_type* “PreAuth”, a pending transaction is created in the system. But later it is converted to Completed by the merchant's confirmation.

Insurance payment:

vpos_type : only applicable for PaySmart2d solution to initiate an insurance payment by an insurance Merchant. If presents value must be “insurance”, otherwise it will throw a validation exception

identity_number : only applicable for PaySmart2d Solution to initiate an insurance payemt by an insurance Merchant when “vpos_type”:”insurance” is present. Value (tckn/vkn/tin) must be within 10-11 digits

cc_no : must be sent in the format : “first_six_digits + ****** + last_six_digits”, i.e. if cc_no is 1234567897891234 then cc_no should be sent like “123456******1234” (please consider the number of asterisks, it should always come as the count of six(6)), otherwise exception will be thrown. Or,

“first_eight_digits + **** + last_four_digits”, i.e. if cc_no is 1234567897891234 then cc_no should be sent like “12345678****1234” (please consider the number of asterisks, it should always come as the count of four(4)), otherwise exception will be thrown.

cvv, expiry_month, expiry_year are optional for Insurance payment

recurring_payment_number

*recurring_payment_number*defines installment count. If first_amount is $100 and **recurring_payment_number**is 5, then the total amount will be deducted as $100*5 = $500. (Cost of transaction may be added with each transaction)

recurring_payment_cycle

recurring_payment_cycle defines the unit type of *recurring_payment_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**: Years

recurring_payment_interval

*recurring_payment_interval*defines interval value. If *recurring_payment_interval*= 2 and *recurring_payment_cycle* = “M” then transaction will occur once in every 2 months.

recurring_web_hook_key

*recurring_web_hook_key*defines merchant recurring web hook url . An URL must be assigned on the Platformode Merchant Panel against this key. Platformode validates this key must exist in the database and it is a required value when payment is recurring.

hash_key

*hash_key*is declared to secure the payment. End user may change the product price before going to the bank. Here is the algorithm to write the hash key given below.

```php
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret){

$data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;

$iv = substr(sha1(mt_rand()), 0, 16);
$password = sha1($app_secret);

$salt = substr(sha1(mt_rand()), 0, 4);
$saltWithPassword = hash('sha256', $password . $salt);

$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);

$msg_encrypted_bundle = "$iv:$salt:$encrypted";
$msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

return $msg_encrypted_bundle;
}
```

Response:

Failed Response:


```
{

    "status_code": 41,

    "status_description": "transaction failed",

    "data": {

        "order_no": "162616264070046",

        "invoice_id": "5edfg345fffgdfdgff",

        "payment_method": 1,

        "credit_card_no": "5355765990527226",

        "transaction_type": "Auth",

        "payment_status": 0,

        "payment_method": 1,

        "error_code": "",

        "error": "transaction failed",

        "hash_key":
"808b7dd19e13efb6:2374:aoDf__tgqgM5fTuFJKFrEcxxC4vTVz2uNarIGX31rzVTP+JZPTteiy5DGam__5wBVk"
```

```
    }
}
```

Success Response:


```
    {
        "status_code": 100,
        "status_description": "Payment process successful",
        "data": {
            "order_no": "162616268649431",
            "invoice_id": "5edfg345fffgdfdgff",
            "payment_method": 1,
            "credit_card_no": "5355765990527226",
            "transaction_type": "Auth",
            "payment_status": 1,
            "payment_method": 1,
            "error_code": 100,
            "error": "Transaction Successful",
            "hash_key":
    "aabb5f5b9f179594:7ade:JiXxJ7wo99Y__9Gm__Kg0lwP273bXxNftgO++vp9ykxyl7n7vmjOKGvtgMkIus8mOk"
    ,
     "merchant_commission": "2.18"
    "user_commission": "0.96",


        }
    }
```

Response Parameter Explanation

Key	Explanation
payment_status	payment_staus can be 1/0. 1=success, 0=fail
order_no	Platformode order no
invoice_id	merchant invoice id
status_code	Platformode Status code, 100 is success code
status_description	Transaction explanation
payment_method	1= Credit Card, 2= Mobile, 3= Wallet
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly. transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	The value of status_code parameter
error	The value of status_description parameter
hash_key	To Validate the request comes from Platformode
merchant_commission	The merchant commission earned from a sale transcation
user_commission	The user commission earned from a sale transaction
settlement_date	Merchant’s settlement date of a transaction

Validate Response:

Condition 1 : status_code== 100 and transaction_type == “Auth” // Transaction is successful and transaction amount is deducted from the card instantly.

Condition 2 : status_code== 100 and transaction_type == “Pre-Authorization” //Process is successful and transaction amount will be deducted from the card later.

11.7. Pay With CardToken

The payByCardToken API is used to submit orders. The user will be redirected to the bank page after the merchant website submits payment form. Payment would be verified by a SMS code at the bank gateway. Once payment is successful, the user will be redirected back to the merchant success URL, otherwise, it will be redirected to cancel the URL set by the merchant. In this payment API, there is no need to call getPos Api like other payment api calls.

Special Notes:

1. **Do not use ajax request to call `payByCardToken`. It must be normal form submission to "`<ACCESS_URL>/api/payByCardToken`"**
2. **Follow sample code given as example file**

Request

Method	URL	Content-Type
POST	`<ACCESS_URL>/api/payByCardToken`	*application/x-www-form-urlencoded*

Type	Params	Data Type	Condition
key	card_token	string	Mandatory
Key	customer_number	number	Mandatory
Key	customer_email	string	Mandatory
Key	customer_phone	string	Mandatory
Key	customer_name	string	Mandatory
Key	customer_phone	string	Mandatory
key	currency_code	string	Mandatory
key	installments_number	number	Mandatory
key	invoice_id	string	Mandatory
key	invoice_description	string	Mandatory

key	total	double	Mandatory
key	merchant_key	string	Mandatory
key	items	string	Mandatory
key	cancel_url	string	Mandatory
key	return_url	string	Mandatory
key	hash_key	string	Mandatory
key	bill_address1	string	Optional
key	bill_address2	string	Optional
key	bill_city	string	Optional
key	bill_postcode	string	Optional
key	bill_state	string	Optional
key	bill_country	string	Optional
key	bill_email	string	Optional
key	bill_phone	string	Optional
key	sale_web_hook_key	string	Optional
key	ip		

Request for Recurring

KEY	order_type	*Integer*	Mandatory
KEY	recurring_payment_number	*Integer*	Mandatory
KEY	recurring_payment_cycle	*string*	Mandatory
KEY	recurring_payment_interval	integer	Mandatory
KEY	recurring_web_hook_key	string	Mandatory
KEY	is_remote_card_token	boolean	Optional (Default: false)

Notes:

cancel_url is used to redirect when payment is failed and return_url is used to redirect when payment is successful.

Regardless of fail or success payment, the following keys are available in the query string with cancel and success url:

payment_status, order_id and invoice_id. For example, if the successs URL is https://<my-domain.com/, the after successful payment, Platformode will redirect to https://<my-domain.com/?payment_status=1&order_no=234234232&invoice_id=73434

In the query string, if the payment_status = 1, the getOrderStatus API must be called to clear cart and change order status to “Completed”

name

name First name of the person. For example, if the name of the person who is buying the product is “john Dao”, then name should be “john”

surname

surname Last name of the person. For example, if the name of the person who is buying the product is “john Dao”, then the surname should be “Dao”.

sale_web_hook_key

*sale_web_hook_key*is an optional key. When a purchase request is completed, Platformode sends a post request. So that merchant can perform an event on their site. Platformode validates that this key must exist in the database. Merchant must assign the Sale web hook URL on the Platformode Merchant Panel against this key.

order_type

If *order_type=1*, Platformode validates payment for recurring. Then *recurring_payment_number, recurring_payment_cycle, recurring_payment_interval*keys should not be empty.

Card_program value must be one of following: "WORLD", "BONUS", "MAXIMUM", "BANKKART_COMBO", "PARAF", "AXESS", "ADVANT", "CARD_FNS"

recurring_payment_number

*recurring_payment_number*defines installment count. If first_amount is $100 and **recurring_payment_number**is 5, then the total amount will be deducted as $100*5 = $500. (Cost of transaction may be added with each transaction)

recurring_payment_cycle

recurring_payment_cycle defines the unit type of *recurring_payment_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**: Years

recurring_payment_interval

*recurring_payment_interval*defines interval value. If *recurring_payment_interval*= 2 and *recurring_payment_cycle* = "M" then transaction will occur once in every 2 months.

recurring_web_hook_key

*recurring_web_hook_key*defines merchant recurring web hook url . An URL must be assigned on the Platformode Merchant Panel against this key. Platformode validates this key must exist in the database and it is a required value when payment is recurring.

is_remote_card_token

is_remote_card_token defines provided *card_token* is generated by remote system or not. For example, if the card token is generated by remote system then the value of *is_remote_card_token* is true else it is false.

hash_key

*hash_key*is declared to secure the payment. End user may change the product price before going to the bank. Here is the algorithm to write the hash key given below.

```
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret){

$data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;

$iv = substr(sha1(mt_rand()), 0, 16);
$password = sha1($app_secret);

$salt = substr(sha1(mt_rand()), 0, 4);
$saltWithPassword = hash('sha256', $password . $salt);

$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);

$msg_encrypted_bundle = "$iv:$salt:$encrypted";
$msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

return $msg_encrypted_bundle;
}
```

Validate Response:

Condition 1 : payment_status == 1 and transaction_type == "Auth" // Transaction is successful and transaction amount is deducted from the card instantly.

Condition 2 : payment_status == 1 and transaction_type == “Pre-Authorization” //Process is successful and transaction amount will be deducted from the card later.

Condition 3 : payment_status == 0 //Transaction is failed

11.8 SaveCard Api

The save Card API is used to store card information to the Platformode System and return a token in the response which will be used in payByCardToken API.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/saveCard	*application/json*

Type	Params	Data Type	Condition
Key	merchant_key	string	Mandatory
Key	card_holder_name	string|max:100	Mandatory
Key	card_number	number	Mandatory
Key	expiry_month	string	Mandatory
Key	expiry_year	string	Mandatory
Key	customer_number	number|unique	Mandatory
Key	hash_key	string	Mandatory
Key	customer_name	string	Optional
Key	customer_phone	string	Optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2Wvzxuxavq
Syu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL
14FSHyeb3lnuF8VFyERwbf-tAll

`Accept`

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

merchant_key

merchant_key is the unique key of the merchant provided by Platformode.

cardholder_name

cardholder_name from card body

card_number

card_number from card body

expiry_month

expiry_month from card body

expiry_year

expiry_year from card body

customer_number

customer_numberfor the customer we will save card, this number must be numeric and unique

customer_name

customer_namefor the customer we will save card

customer_phone

customer_phonefor the customer we will save card

hash_key

hash_key is declared to secure the save card api request. End user may change data before going to Platformode api for save card. Here is the algorithm to write the hash key given below.

```
function generateHashKey($merchant_key, $customer_number, $card_number,
$card_holder_name, $customer_number, $expiry_month, $expiry_year, $app_secret){


$data = $merchant_key.'|'. $customer_number
'|'.$card_holder_name.'|'.$card_number.'|'.$expiry_month.'|'.$expiry_year;


$iv = substr(sha1(mt_rand()), 0, 16);

$password = sha1($app_secret);


$salt = substr(sha1(mt_rand()), 0, 4);

$saltWithPassword = hash('sha256', $password . $salt);


$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);


$msg_encrypted_bundle = "$iv:$salt:$encrypted";

$msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);


return $msg_encrypted_bundle;

}
```

Response:

Failed Response:

```
{
   "status_code": "86",
"status_description": "The card token save failed"
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "The card token saved successfully",
    "card_token": "WNPDZDQNMVNRQO23IAHKDKXIWCRGWHEXCFNRDXXK5CMXGM5A"
}
```

11.9 Save Card Edit Api

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/editCard	*application/json*

Type	Params	Data Type	Condition
Key	merchant_key	string	Mandatory
Key	card_token	string	Mandatory
Key	customer_number	string	Mandatory
Key	expiry_month	string	Mandatory
Key	expiry_year	string	Mandatory
Key	hash_key	string	Mandatory
Key	card_holder_name	string|max:100	Optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2WvzxuxavqSyu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL14FSHyeb3lnuF8VFyERwbf-tAll

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

merchant_key

merchant_key is the unique key of the merchant provided by Platformode.

card_token

card_tokensaved at Platformode system

expiry_month

expiry_month from card body

expiry_year

expiry_year from card body

customer_number

customer_number for the customer we will save card, this number must be numeric and unique

hash_key

*hash_key*is declared to secure the save card api request. End user may change data before going to Platformode api for save card. Here is the algorithm to write the hash key given below.

```
function generateHashKey($merchant_key, $customer_number, $card_token,  $app_secret){

        $data = $merchant_key.'|'.$customer_number.'|'.$card_token

        $iv = substr(sha1(mt_rand()), 0, 16);

        $password = sha1($app_secret);

        $salt = substr(sha1(mt_rand()), 0, 4);

        $saltWithPassword = hash('sha256', $password . $salt);

        $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null,
        $iv);

        $msg_encrypted_bundle = "$iv:$salt:$encrypted";

        $msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

        return $msg_encrypted_bundle;

}
```

Response:

Failed Response:

```
{
   "status_code": 88,
    "status_description": "The card token deletion failed",
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "The card token updated successfully",
```

```
"card_token": "WNPDZDQNMVNRQO23IAHKDKXIWCRGWHEXCFNRDXXK5CMXGM5A"
}
```


11.10 SaveCard Delete Api

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/deleteCard	*application/json*

Type	Params	Data Type	Condition
Key	merchant_key	string	Mandatory
Key	card_token	string	Mandatory
Key	customer_number	string	Mandatory
Key	hash_key	string	Mandatory

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM

GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2Wvzxuxavq
Syu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL
14FSHyeb3lnuF8VFyERwbf-tAll

`Accept`

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

card_token

card_tokensaved at Platformode system

customer_number

customer_number for the customer we will save card, this number must be numeric and unique

hash_key

hash_key is declared to secure the save card api request. End users may change data before going to Platformode api for save card. Here is the algorithm to write the hash key given below.

```
function generateHashKey($merchant_key, $customer_number, $card_token,  $app_secret){

        $data = $merchant_key.'|'.$customer_number.'|'.$card_token

        $iv = substr(sha1(mt_rand()), 0, 16);

        $password = sha1($app_secret);

        $salt = substr(sha1(mt_rand()), 0, 4);

        $saltWithPassword = hash('sha256', $password . $salt);
```

```
        $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null,
        $iv);

        $msg_encrypted_bundle = "$iv:$salt:$encrypted";

        $msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

        return $msg_encrypted_bundle;

}
```

Response:

Failed Response:

```
{
   "status_code": 41,
    "status_description": "Invalid merchant Key",
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "Card deleted successfully",
}
```

11.11 SaveCard : Get Card Tokens Api

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/getCardTokens	*application/json*

Type	Params	Data Type	Condition
Key	merchant_key	string	Mandatory
Key	customer_number	string	Mandatory

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-
Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO28luCMOFf6UHn6XxeEhK3XWaboZJvrubdeb0t04a6btrdHUa-
FgeV6I8bNSRlzUjOjBcsVrd1pxKhKnsREFHCWfzYVC_ZQ4RRCi9CZsJGz7_KQ8mo0BdNmtbNKwfvYkpcdsmVicsJYvn
w7OMZ3u-TorhakndhQkUK0JPAzl_LSHqAKCju8dTG1-
vZjbh9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-
cmjDa6aeGNjZHfNa7dQ8bTokdbkxqwKrVVUUVjgkMtPXhpL9yffaHHPNBCkc-
1Vz40nsmNFeaoWlk2S7fDxFTcGYv8HFFiSRyfsPpfTbXPIRMoZUX1kC4c-
DMyQmjuBqtxIwEFzJexs9PkZEUze5Qcm_ZrkqeKUlL4tJidO9ZzwfCI9bpihMATHlDyM6IP7XyhgMRt3yr2Wvzxuxavq
Syu09YlybYU0WpTUtDVOavL7xnuKBXhwDSoCjtCMh__tL9ZfK9lDvq6mrHQ5Z4RXLixvWMbl98_Btbnfg_SqnCNYwL
14FSHyeb3lnuF8VFyERwbf-tAll

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

merchant_key

merchant_key is the unique key of the merchant provided by Platformode .

customer_number

customer_number for the customer we will save card, this number must be numeric and unique

Response:

Failed Response:

{

```
   "status_code": 41,
    "status_description": "Invalid merchant Key",
}
```


Success Response:


```
{
    "status_code": 100,
    "status_description": "Data fetched successfully",
    "data": [
        {
            "id": 11,
            "pos_id": 0,
            "card_token": "KELUUKPQBCGYLV7FP7I7VUQNZP7N5UX7JUCPKZVOPKEMCEQP",
            "merchant_id": 98950,
            "customer_number": "1122343443-98950",
            "customer_name": "Taygun Alban",
            "customer_email": "taigun@gmail.com",
            "customer_phone": "+8801749452019",
            "bin": "540668",
            "created_at": "2021-04-08T14:07:42.000000Z",
            "updated_at": "2021-04-08T14:07:42.000000Z"
        }
    ]
}
```


11.12 Get Transactions

The API can be used for getting sale transactions by specific date.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/getTransactions	*application/json*

Type	Params	Data Type	Condition

HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
HEADER	Content-Type	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	hash_key	string	Mandatory
KEY	date	string	Optional
KEY	invoiceid	string	Optional
KEY	currency_id	int	Optional
KEY	paymentmethodid	int	Optional
KEY	minamount	int	Optional
KEY	maxamount	int	Optional
KEY	transactionState	int	Optional
KEY	type	string	Optional
KEY	order_by_option	int	Optional
KEY	from_date	datetime	Optional
KEY	to_date	datetime	Optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6lPoa1rfG
3M4jfsNeH-Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

`Content-Type`

Content-Typedetermines what type of representation is desired at server side. The value should be "*application/json*"

merchant_key

merchant_key merchant_key value of the auth merchant

hash_key

hash_key hash_key is generated using an specific algorithm that we have provided below-

```
function generateAPIHashKey($merchant_key, $date, invoiceid, $currency_id, $paymentmethodid,
$minamount, $maxamount, $transactionState){

    $data =
$date.'|'.invoiceid.'|'.$currency_id.'|'.$paymentmethodid.'|'.$minamount.'|'.$maxamount.'|'.$trans
actionState;

    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($merchant_key);

    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);

    $encrypted = openssl_encrypt(
"$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
    );
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $hash_key = str_replace('/', '__', $msg_encrypted_bundle);
return $hash_key;
}
```

date

if the **type** field is empty or is not empty, but the value of the type field is "daily" then the date field is required and the valid format is '2021-04-20'.

if the **type** field is not empty and the value of the type field is "hourly" then the date field is required and the valid format is '2021-04-20 21:00:00'.

if the **type** field is not empty and the value of the type field is "range" then the date field is optional.

invoiceid

string value, nullable.

currency_id

int value, nullable.

paymentmethodid

int value, nullable.

minamount

int value, nullable.

maxamount

int value, nullable.

transactionState

int value, nullable.

type

string constant value. Value of this field can be, “daily” or “hourly” or “range”. If the type field is not present in the request body, then the system will consider as “daily”

order_by_option

Nullable. Enum : as described below

1 (default) => range query will look through creation date using “*from_date*” and “*to_date*” .

2 => range query will look through modification date using “*from_date*” and “*to_date*”

from_date

This is a datetime field. If the value of the type field is “range” then the from_date field is required. Otherwise it’s optional. Valid Datetime format: 2021-04-20 21:00:00
Can be used for date range wise filtering.

to_date

This is a datetime field. If the value of the type field is “range” then the from_date field is required. Otherwise it’s optional. Valid Datetime format: 2021-04-20 21:00:00
Can be used for date range wise filtering.

Post Request Params:


```
{
   "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he",
   "hash_key": "823e40a698c3ca1f:4c80:cYTC/9s5ehGRtGeYyXA0Tw==",
   "date": "2021-02-10",
   "invoiceid": "",
   "currency_id": "",
   "paymentmethodid": "",
   "minamount": "",
   "maxamount": "",
   "transactionState": "",
   “type”: daily
}
```

Responses:

Failed Response:

```
{
   "status_code": 90,
   ""status_description": "Invalid Hash key"
}
```

Success Response:

```
    {
        "status_code": 100,
        ""status_description": "Success",
        "data": [
            {
                "id": 275018,
                "user_id": 12,
                "merchant_id": 98950,
                "payment_id": "W7iQ8-M7jb-TC10-98950-100221",
                "transaction_state_id": 11,
                "merchant_commission": 0.665,
                "gross": 11.3,
                "net": 10.635,
                "payment_type_id": 1,
                "product_price": 10,
```

"total_refunded_amount": 10,

"card_holder_bank": "T. VAKIFLAR BANKASI T.A.O.",

"operator": "",

"user_name": "",

"dpl_id": 0,

"currency_id": 1,

"invoice_id": "3905388578454743445487",

*"credit_card_no": "535576****7226",*

"installment": 1,

"rolling_amount": 1.13,

"gsm_number": "",

"payment_source": 12,

"card_program": null,

"result": "Approved(21041MekI07029311)",

"settlement_date_merchant": "2021-02-11 14:00:00",

"created_at": "2021-02-10T12:30:37.000000Z",

"updated_at": "2021-03-09T15:14:27.000000Z",

"recurring_id": 0,

"order_id": "16129494361217",

"ip": "103.195.141.79",

"auth_code": "175705",

"card_issuer_name": null,

"refunded_chargeback_fee": 50,

"pos_name": "ISBANK - Günlük TRY MAXIMUM",

"settlement_date_bank": "2021-02-11 14:00:00",

"transaction_state_label": "Chargebacked",

"payment_type_label": "MP Credit Card",

```
"currency_code": "TRY",
"rolling_balance": {
    "id": 10847,
    "sale_id": 275018,
    "effective_date": "2021-08-10 11:00:00"
},
"sale_billing": {
    "id": 15043,
    "sale_id": 275018,
    "card_holder_name": "A******"
},
"merchant_sale": {
    "id": 161297,
    "sale_id": 275018,
    "merchant_commission_percentage": 5,
    "merchant_commission_fixed": 0.1,
    "end_user_commission_percentage": 3,
    "end_user_commission_fixed": 1,
    "merchant_rolling_percentage": 10,
    "cot_percentage": 0.75,
    "cot_fixed": 0
},
"refund_history": [],
"sale_recurring": null,
"sale_recurring_history": [],
"sale_integrator": {
    "id": 2100,
```

```
        "sale_id": 275018,
        "integrator_id": 3,
        "commission_percentage": 0.1,
        "commission_fixed": 0,
        "commission_amount": 0.0113,
        "amount": 11.3,
        "integratordata": {
            "id": 3,
            "integrator_name": "Aigerim integrator",
            "status": 1
        }
      },
      "userdata": {
        "id": 12,
        "name": "Test qqqqa"
      },
      "merchant": {
        "user_id": 12,
        "name": "Platformode ",
        "mcc": "6012"
      }
    }
  ]
}
```


11.13 Payment Confirmation **API**

In PreAuth Payment, a transaction is created as Pending in the system. To make this transaction Failed or Completed, this API should be called using request parameters given below.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/confirmPayment	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	invoice_id	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	status	integer	Mandatory
KEY	hash_key	string	Mandatory
KEY	total	double	Optional

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3ViIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6lPoa1rfG3M4jfsNeH

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

invoice_id

invoice_id is a unique order id sent by merchant

merchant_key

merchant_key is the unique key of the merchant provided by Platformode.

status

status can be 1 or 2 (1=approve, 2 = Cancel)

hash_key

hash_key is a unique key that should be generated using input value. The method for generating hash_key given below

```
function generateConfrimPaymentHashKey($merchant_key, $invoice_id,
$status, $app_secret)
{

   $data = $merchant_key . '|' . $invoice_id . '|' . $status;

   $iv = substr(sha1(mt_rand()), 0, 16);
   $password = sha1($app_secret);

   $salt = substr(sha1(mt_rand()), 0, 4);
   $saltWithPassword = hash('sha256', $password . $salt);

   $encrypted = openssl_encrypt(
     "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
   );
   $msg_encrypted_bundle = "$iv:$salt:$encrypted";
   $msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);
   return $msg_encrypted_bundle;
}
```

total
the value of *total* parameter must be less than or equal to product price of the pre auth transaction

Response:

Failed Response:

```
{
    "status_code": 105,
    "status_description": " The transaction is not Approved",
    "transaction_status": "Pending",
    "order_id": "162435924998223",
    "invoice_id": "50781624359247
```

```
}
```

Success Response:

```
{
    "status_code": 100,
    "status_description": "An order has been taken place for this invoice id: 33491162435928”,
    "transaction_status": "Completed",
    "order_id": "162435932934307",
    "invoice_id": "33491162435928"

}
```

11.14. Pay With Card Token (Non-Secure)

The payByCardToken(non-secure) API is used to submit orders to the Platformode payment integration system. Merchant websites should receive payment status immediately without loading the checkout page. Based on API success status, cart and order status must be changed accordingly. In this payment API, there is no need to call getPos Api like other payment API calls.

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/payByCardTokenNonSecure	*application/json*

Type	Params	Data Type	Condition
key	card_token	string	Mandatory
Key	customer_number	number	Mandatory
Key	customer_email	string	Mandatory
Key	customer_phone	string	Mandatory
Key	customer_name	string	Mandatory
Key	customer_phone	string	Mandatory
key	currency_code	string	Mandatory

key	installments_number	number	Mandatory
key	invoice_id	string	Mandatory
key	invoice_description	string	Mandatory
key	total	double	Mandatory
key	merchant_key	string	Mandatory
key	items	string	Mandatory
key	hash_key	string	Mandatory
key	bill_address1	string	Optional
key	bill_address2	string	Optional
key	bill_city	string	Optional
key	bill_postcode	string	Optional
key	bill_state	string	Optional
key	bill_country	string	Optional
key	bill_email	string	Optional
key	bill_phone	string	Optional
key	sale_web_hook_key	string	Optional
key	ip		

Request for Recurring

KEY	order_type	*Integer*	Mandatory
KEY	recurring_payment_number	*Integer*	Mandatory
KEY	recurring_payment_cycle	*string*	Mandatory
KEY	recurring_payment_interval	integer	Mandatory
KEY	recurring_web_hook_key	string	Mandatory
KEY	is_remote_card_token	boolean	Optional (Default: false)

Notes:

name

name First name of the person. For example, if the name of the person who is buying the product is “john Dao”, then name should be “john”

surname

surname Last name of the person. For example, if the name of the person who is buying the product is “john Dao”, then the surname should be “Dao”.

sale_web_hook_key

*sale_web_hook_key*is an optional key. When a purchase request is completed, Platformodesends a post request. So that merchant can perform an event on their site. Platformodevalidates that this key must exist in the database. Merchant must assign the Sale web hook URL on the PlatformodeMerchant Panel against this key.

order_type

If *order_type=1*, Platformodevalidates payment for recurring. Then *recurring_payment_number, recurring_payment_cycle, recurring_payment_interval*keys should not be empty.

transaction_type

For *transaction_type* “PreAuth”, a pending transaction is created in the system. But later it is converted to Completed by the merchant's confirmation.

recurring_payment_number

*recurring_payment_number*defines installment count. If first_amount is $100 and **recurring_payment_number**is 5, then the total amount will be deducted as $100*5 = $500. (Cost of transaction may be added with each transaction)

recurring_payment_cycle

recurring_payment_cycle defines the unit type of *recurring_payment_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**: Years

recurring_payment_interval

*recurring_payment_interval*defines interval value. If *recurring_payment_interval*= 2 and *recurring_payment_cycle* = “M” then transaction will occur once in every 2 months.

recurring_web_hook_key

*recurring_web_hook_key*defines merchant recurring web hook url . An URL must be assigned on the Platformode Merchant Panel against this key. Platformode validates this key must exist in the database and it is a required value when payment is recurring.

is_remote_card_token

is_remote_card_token defines provided *card_token* is generated by remote system or not. For example, if the card token is generated by remote system then the value of *is_remote_card_token* is true else it is false.

hash_key

*hash_key*is declared to secure the payment. End user may change the product price before going to the bank. Here is the algorithm to write the hash key given below.

```
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret){

$data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;

$iv = substr(sha1(mt_rand()), 0, 16);
$password = sha1($app_secret);

$salt = substr(sha1(mt_rand()), 0, 4);
$saltWithPassword = hash('sha256', $password . $salt);

$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);

$msg_encrypted_bundle = "$iv:$salt:$encrypted";
$msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);

return $msg_encrypted_bundle;
}
```

Response:

Failed Response:


```
{

    "status_code": 41,

    "status_description": "transaction failed",

    "data": {

        "order_no": "162616264070046",
```

```
        "invoice_id": "5edfg345fffgdfdgff",
        "payment_method": 1,
        "credit_card_no": "5355765990527226",
        "transaction_type": "Auth",
        "payment_status": 0,
        "payment_method": 1,
        "error_code": "",
        "error": "transaction failed",
        "hash_key":
"808b7dd19e13efb6:2374:aoDf__tgqgM5fTuFJKFrEcxxC4vTVz2uNarIGX31rzVTP+JZPTteiy5DGam__5wBVk"
    }
}
```


Success Response:


```
    {
        "status_code": 100,
        "status_description": "Payment process successful",
        "data": {
            "order_no": "162616268649431",
            "invoice_id": "5edfg345fffgdfdgff",
            "payment_method": 1,
            "credit_card_no": "5355765990527226",
            "transaction_type": "Auth",
            "payment_status": 1,
            "payment_method": 1,
```

```
        "error_code": 100,

        "error": "Transaction Successful",

        "hash_key":
"aabb5f5b9f179594:7ade:JiXxJ7wo99Y__9Gm__Kg0lwP273bXxNftgO++vp9ykxyl7n7vmjOKGvtgMkIus8mOk"

    }

}
```


Response Parameter Explanation

Key	**Explanation**
payment_status	payment_staus can be 1/0. 1=success, 0=fail
order_no	Platformode order no
invoice_id	merchant invoice id
status_code	Platformode Status code, 100 is success code
status_description	Transaction explanation
payment_method	1= Credit Card, 2= Mobile, 3= Wallet
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly. transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	The value of status_code parameter
error	The value of status_description parameter
hash_key	To Validate the request comes from Platformode

Validate Response:

Condition 1 : status_code== 100 and transaction_type == “Auth” // Transaction is successful and transaction amount is deducted from the card instantly.

Condition 2 : status_code== 100 and transaction_type == “Pre-Authorization” //Process is successful and transaction amount will be deducted from the card later.

11.15. All Transaction API

The API can be used for getting sale transactions by provided request params

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/alltransaction	*application/json*

INPUT Parameters for the request JSON

Type	Params	Data Type	Condition
Header	Header/Authorization	Bearer/string	Mandatory
key	transaction_id	string	Optional
key	invoice_id	string	Optional
key	order_id	string	Optional
key	payment_method	int	Optional
key	transactionState	string	Optional
key	date_range	string	Optional
key	currency_code	string	Optional
key	amount	integer	Optional

key	maxamount	integer	Optional
key	minamount	integer	Optional
key	billing_phone	string	Optional
key	billing_email	string	Optional
key	search_key	any	Optional
key	order	string	Optional
key	page_limit	string	Optional

Params Description:

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be “Bearer”

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V
iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG
3M4jfsNeH-Sh7g6PaVffIoKvjdcUG7Cc2lLqhE4qMEdPgO

`Accept`

Accept determines what type of representation is desired at client side. The value should be “*application/json*”

`Content-Type`

Content-Typedetermines what type of representation is desired at server side. The value should be “*application/json*”

transaction_id
Sales transaction id . example ‘vPiWL-pMmb-TC10-98950-230122’

invoice_id
Sales invoice id. Example ‘1234’

order_id
Sales order id Example ‘164290959020713

payment_method
Paymet method. `1= Credit Card, 2= Mobile, 3= Wallet`

`date_range`

date format should be like example- ‘2021-04-20 - 2021-04-20’ etc.

`currency_code`

string value, By default, it is set as “TRY”. Platformode Only allows “TRY”, “USD”, “EUR”

`amount`

int value, nullable. `‘10.00’`

`minamount`

int value, nullable.

`maxamount`

int value, nullable.

`billing_phone`

string value, nullable.

`billing_email`

string value, nullable.

`transactionState`

int value, nullable. 1= Completed, 3 = Pending..

search_key

int,string value. Search by some column name, like order_id, invoice_id, merchant_name etc

order

string value , asc/desc

page_limit

int/string value 10,20 ..

Headers

Authorization: Bearer
Content-Type: application/json

Request


```
{
   "transaction_id": "",
   "invoice_id": "",
   "order_id": "",
   "payment_method": "",
   "transactionState": "",
   "date_range": "2021/12/23 - 2022/01/23",
   "currency_code": "TRY",
   "amount": "",
   "maxamount": "",
   "minamount": "",
   "billing_phone": "",
   "billing_email": "",
   "search_key": "vPiWL-pMmb-TC10-98950-230122",
   "order": "desc",
   "page_limit":"2"
}
```


Response

Status	Response

200	**Successful**
	{
	"status_code": 100,
	"status_description": "data fetched successfully",
	"data": {
	"current_page": 1,
	"data": [
	{
	"id": 287447,
	"payment_id": "vPiWL-pMmb-TC10-98950-230122",
	"invoice_id": "14391",
	"order_id": "164290959020713",
	"remote_order_id": "",
	"user_id": 12,
	"end_user_id": 0,
	"user_name": "",
	"merchant_id": 98950,
	"merchant_name": "Fintlix Demo",
	"transaction_state_id": 3,
	"purchase_id": 0,
	"gross": 5,

```
            "refund_request_amount": 0,
            "fee": 0.1,
            "net": 4.9,
            "pay_by_token_fee": 0,
            "cost": 0.05,
            "rolling_amount": 0,
            "product_price": 5,
            "user_commission": 0,
            "merchant_commission": 0.1,
            "gsm_number": "",
            "settlement_date_bank": "",
            "settlement_date_merchant": "",
            "payment_type_id": 1,
            "operator": "",
            "issuer_name": "DENIZBANK A.S.",
            "card_program": "",
            "card_issuer_name": "",
            "pos_name": "Deniz DCC Try Pos",
            "pos_id": 113,
            "allocation_id": 0,
            "campaign_id": 0,
            "dpl_id": 0,
            "json_data": "{\"action\":\"WL_REQUEST_PAY_SMART_3D\",\"transaction_type\":\"Auth\",\"merchant_key\":\"$2y$10$w\\/ODdbTmfubcbUCUq\\/ia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he\",\"invoice_id\":\"14391\",\"total\":\"5\",\"items\":[{\"name\":\"Test Product\",\"price\":\"5.00\",\"quantity\":1,\"description\":\"Some Description\"}],\"currency_code\":\"TRY\",\"cc_no\":\"cJ>_uWL@_fv<2ad9:tE@_:bxl37OOAv0mU4GU4PngAsnLel6pIezOQR\\/kZtJeryKE=\",\"installments_number\":1,\"cancel_url\":\"https:\\/\\/localhost\\/app.Platformode.com.tr\\/api\\/payment_success.php?merchant_key=%242y%2410%24w%2FODdbTmfubcbUCUq%2Fia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he\",\"return_url\":\"https:\\/\\/localhost\\/app.Platformode.com.tr\\/api\\/payment_success.php?merchant_key=%242y%2410%24w%2FODdbTmfubcbUCUq%2Fia3OoJFMUmkM1UVNBiIQIuLfUlPmaLUT1he\",\"hash_key\":\"645e82f0848a9279:e319:x2odQXfr+xcmL0OSE6qMxs81SJiAu9p+nQjaY618glpH5Io6+4pFL68rNZYZkz+B4l4+l3JpF2zIEGG7k8pM63G0+jbsct+gEVv3cbjNEvg=\",\"name\":\"Customer Name\",\"surname\":\"Customer surname\",\"saved_card\":0,\"maturity_period\":\"0\",\"payment_frequency\":\"0\",\"pf_id\":\"\",\"bill_address1\":\"Address 1 should not more than 100\",\"bill_address2\":\"Address 2\",\"bill_city\":\"Istanbul\",\"bill_postcode\":\"1111\",\"bill_state\":\"Istanbul\",\"bill_country\":\"TURKEY\",\"bill_phone\":\"008801777711111\",\"bill_email\":\"demo@your-domain.com\",\"sale_web_hook_key\":\"\",\"merchant_server_id\":\"\",\"referer_url\":\"https:\\/\\/localhost\\/app.Platformode.com.tr\\/public\\/api\\/paySmart3D\\/payment_page.php\",\"api_name\":\"PAY_SMART_3D\",\"payment_type\":\"3d\",\"is_pay_by_marketplace\":0,\"currency_id\":1,\"pos_id\":113,\"campaign_id\":0,\"allocation_id\":0,\"is_tarim_payment\":false,\"installment\":1,\"merchant_ip\":\"\",\"part_2\":\"cJ>_uWL@_fv<2ad9:tE@_:jocdjiBevAneTYND6J6mOg==\",\"part_1\":\"cJ>_uWL@_fv<2ad9:tE@_:oFAEwm\\/gdXomWrR\\/QofemQ==\"}",
            "document": "",
            "total_refunded_amount": 0,
            "refund_reason": "",
            "refund_request_date": "",
            "admin_process_date": "",
            "chargeback_reject_explanation": "",
            "created_at": "2022-01-23T03:46:30.000000Z",
            "updated_at": "2022-01-23T03:46:30.000000Z",
            "currency_id": 1,
            "currency_symbol": "₺",
            "installment": 1,
            "maturity_period": 0,
```

```
            "payment_frequency": 0,
            "ip": "::1",
            "result": "",
            "credit_card_no": "546938****2434",
            "card_holder_bank": "",
            "sale_version": 0,
            "sale_type": 1,
            "payment_source": 5,
            "refunded_chargeback_fee": 0,
            "is_bank_refund_failed": 0,
            "recurring_id": 0,
            "sale_web_hook_key": "",
            "auth_code": "",
            "migration_status": 3,
            "is_comission_from_user": 0,
            "sale_recurring": "",
            "paymenttype": {
              "id": 1,
              "name": "Credit Card",
              "created_at": "2019-07-31T02:52:28.000000Z",
              "updated_at": "2019-07-31T02:52:28.000000Z"
            },
            "rolling_balance": "",
            "sale_integrator": "",
            "sale_billing": "",
            "sale_property": ""
          }
      ],
      "first_page_url": "https://localhost/app.Platformode.com.tr/ccpayment/api/alltransaction?page=1",
      "from": 1,
      "last_page": 1,
      "last_page_url": "https://localhost/app.Platformode.com.tr/ccpayment/api/alltransaction?page=1",
      "next_page_url": "",
      "path": "https://localhost/app.Platformode.com.tr/ccpayment/api/alltransaction",
      "per_page": "2",
      "prev_page_url": "",
      "to": 1,
      "total": 1
    }
}
```


200

Failure


```
{

   "status_code": 99
   "status_description": "failed",
   "data": []
}
```

11.16 Merchant 3D Model : CompletePayment Api

This is the payment capturization api where payment acquisition step can be done by merchant after a transaction being initiated with "*payment_completed_by*" : "merchant" and "md_status" : 1

Request

Method	URL	Content-Type
POST	<ACCESS_URL>/payment/complete	*application/json*

Type	Params	Data Type	Condition
HEADER	Authorization	string	Mandatory
HEADER	Accept	string	Mandatory
KEY	invoice_id	string	Mandatory
KEY	merchant_key	string	Mandatory
KEY	order_id	string	Mandatory
KEY	status	string	Mandatory
KEY	hash_key	string	Mandatory

`Authorization`

Authorization is a header key which defines verification that the connection attempt is allowed. The method should be "Bearer"

Example value:

Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTU
xYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkyMjdlZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlM
GVlZGFiZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTky
MjdlZGVmZDdlMDliIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOjE1NzM3NTI0NzIsImV4cCI6MTYwNTM3NDg3Miwic3V

iIjoiMSIsInNjb3BlcyI6W119.mDtdzcv15p8SnYjZYJUJrhdskO5NohXbkcAxKWWZ72lNtrg86RZ1yxQwfQlRu6IPoa1rfG3M4jfsNeH

Accept

Accept determines what type of representation is desired at client side. The value should be "*application/json*"

invoice_id

invoice_id is a unique order id sent by merchant

merchant_key

merchant_key is the unique key of the merchant provided by brand.

status

status can be "complete" or "cancel" (complete will initiate the acquisition step where cancel will diretly initiate the payment cancellation stage)

hash_key

hash_key is a unique key that should be generated using input value. The method for generating hash_key given below

```
function generateConfrimPaymentHashKey($merchant_key, $invoice_id,
order_id,  status, app_secret)
{

   $data = $merchant_key . '|' . $invoice_id . '|' . order_id . '|' .
$status;

   $iv = substr(sha1(mt_rand()), 0, 16);
   $password = sha1($app_secret);

   $salt = substr(sha1(mt_rand()), 0, 4);
   $saltWithPassword = hash('sha256', $password . $salt);

   $encrypted = openssl_encrypt(
     "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
   );
   $msg_encrypted_bundle = "$iv:$salt:$encrypted";
   $msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);
   return $msg_encrypted_bundle;
}
```

Request:

{

"merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBiIQluLfUlPmaLUT1he",

"invoice_id": "9a197ed9-cfaf-4456-b0fc-da74a95f461c",

"order_id": "166254527653758",

"status": "complete",

"hash_key":
"36a50210c8b2f19e:11629:eHBA0moN6bZAOSfVC6g9uapUAt1/zxnxko9x4uP73/TsqrQM+cCYSekMi/VnqMgjx3GpZLp
Bu5GW1BDIOyxigKz3oqBFQwItyA31S5bhHe5Q3rWLe9WBStaiDnFbWYfWbBL8p7BKIbKZKDCtDHQ41B+4PEIPhdpNCei
hKmE2fto="

}

Response:

Failed Response:

{

"status_code": 31,

"status_description": "Pending sale not found against order_id 166246147223278",

"data": {

"invoice_id": "f6a2a107-088f-4169-875f-8a6f30cf4ff2",

"order_id": "166246147223278"

}

}

Success Response:

{

"status_code": 100,

"status_description": "Payment Successfully Completed",

"data": {

"order_no": "166254527653758",

"order_id": "166254527653758",

"invoice_id": "9a197ed9-cfaf-4456-b0fc-da74a95f461c",

"status_code": 100,

"status_description": "Payment Successfully Completed",

"credit_card_no": "557023****7463",

"transaction_type": "Auth",

"payment_status": **true**,

"payment_method": 1,

"error_code": 100,

"error": "Payment Successfully Completed",

"auth_code": "287679",

"status": "Completed",

"hash_key":
"c98d63c4ba1952c9:19e8:94Z0B48k__V5pCWIXlov3xbDKmM8beaNE3Lk4v3Yu5A1wfdPulFnDsJgFKJX+2gOqA2AfjcRExhYIuWCF3A4LireIUkv26kozmIHc0gjbwIY="

}

}

Important Note:

For response validation for payment please see the response section of Pay With White Label Smart 3D API section.

12.0 Server Status Codes

Status Code	Description
200	OK
201	Created

202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

13.0 Application Status Codes

Status Code	Description
1	Basic Validation
3	The invoice id already processed, The order with this invoice id 455323123214 is in the process, please wait or create order with new invoice id

12	Items must be an array, Invalid Currency code, Invalid format of items
13	The total of your item's price is not equal to the invoice total.
14	Merchant Not Found !
30	Invalid credentials
31	Transaction Not Found
32	Invalid Invoice Id, Order Not Completed
33	Quantity must be integer value
34	The payment integration method is not allowed. Please contact support.
35	Credit Card Payment Option has not been defined
36	POS Not Found
37	Merchant Pos Commission was not set. Please contact with service provider
38	The Merchant Commission was not set for this currency and payment method. Please try another payment method
39	commission not found
40	Installment not found
41	Order Failed
41	Payment Failed
42	Product price is less than commission, Product price is less than cost
43	Settlement date was not set
44	This credit card is blocked.
44	Merchant daily number of transaction limit is crossed
45	Merchant daily transaction amount limit is crossed

46	Merchant monthly number of transaction limit is crossed
47	Merchant monthly transaction amount limit is crossed
48	POS currency does not matches with expected currency id, Minimum transaction limit per transaction has been violated, Minimum transaction limit per transaction has been violated, POS currency does not matches with expected currency id
49	Refund Failed, Total refund amount should not cross net amount, Sale Transaction not found, Insufficient balance, Non refund Transaction State, Please wait at least 30 seconds for another refund, Maximum transaction limit per transaction has been violated
55	Recurring payment can not be installment sale., Recurring number can not be empty, Recurring number must be an integer, Recurring number must be greater than 1, Recurring number should not be greater than 121, The recurring cycle can not be empty, The recurring cycle unit is not valid. It should be 'D', 'M' or 'Y', Recurring interval can not be empty, Recurring interval must be an integer, Recurring interval must be greater than 0, Recurring interval should not be greater than 99, Invalid recurring web hook key! Please check key name on Platformode , Recurring web hook key can not be empty. Please assign your web hook key on Platformode ,
56	Invalid sale web hook key! Please check key name on Platformode
60	The merchant is not allowed to perform transactions using this card.
68	Total Amount mismatch with hash key,

	Currency mismatch with hash_key, Merchant key mismatch with hash key, Installment number mismatch with hash key, Pos Id mismatch with hash key, Invoice id mismatch with hash key
69	Order is not processed yet
70	Card Program Mismatch, Recurring Not Found
71	Recurring plan update failed, Invalid Response or Unknown Error
72	Failed to delete, Old payment could not be processed
73	Recurring plan card add failed, Invalid Response or Unknown Error
76	Foreign Cards are not Allowed for this Merchant.
76	Foreign Cards are not Allowed for this Merchant.
77	Foreign Card Commission is not set for this merchant
79	Merchant is not allowed to pay with card token
80	Sub Merchant not found
81	Currency conversion failed from API
85	invalid character
86	Save Card Token failed
87	Invalid Token or customer number
88	The card token deletion failed
89	The card token update failed
90	Invalid Hash key
91	Hash key mismatched with merchant key
92	Hash key mismatched with customer number
93	Hash key mismatched with card holder name

94	Hash key mismatched with card number
95	Hash key mismatched with expiry month
96	Hash key mismatched with expiry year
97	Hash key mismatched with card token
99	Unknown Error
100	Success
102	Pos was not defined, Pos not found for this installment
103	PayBytoken commission was no setup
104	Refund transaction id must be unique
105	Transaction is not approved
106	Invalid Merchant Type
107	Sent API OTP Unverified
108	Invalid Card Number
109	File Processing Error
110	Partial Refund is not allowed for this transaction
112	Partially Successful
404	Cancel Url should not be empty
113	There is no pre auth transaction found for this operation